

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky  
a komunikačních technologií

DIPLOMOVÁ PRÁCE



# VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

## FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

## ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

## MONITOROVÁNÍ VÝROBNÍCH ZAŘÍZENÍ

MONITORING OF PRODUCTION FACILITIES

### DIPLOMOVÁ PRÁCE

MASTER'S THESIS

### AUTOR PRÁCE

AUTHOR

Bc. Adam Borsuk

### VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. Petr Číka, Ph.D.

BRNO 2020

# Diplomová práce

magisterský navazující studijní obor **Telekomunikační a informační technika**

Ústav telekomunikací

**Student:** Bc. Adam Borsuk

**ID:** 173616

**Ročník:** 2

**Akademický rok:** 2019/20

**NÁZEV TÉMATU:**

## Monitorování výrobních zařízení

### POKYNY PRO VYPRACOVÁNÍ:

Cílem práce je navrhnout hardwarové (HW) zařízení, které umožní centralizovaný sběr dat informujících o vytížení jednotlivých strojů ve výrobě, o době jejich běhu, prostojích či poruchách tak, aby je bylo možné efektivně vytěžovat. Práce bude obsahovat mimo návrhu HW zařízení taktéž jeho realizaci na nepájivém poli minimálně s následující moduly:

1. přihlášení a verifikaci zaměstnance přes RFID nebo jiným způsobem,
2. WiFi připojení do firemní sítě,
3. zobrazení aktuálního stavu stroje a jiných počítadel - TFT / LED displej,
4. snímání s minimálně osmi digitálními vstupy / výstupy - odděleně pro signály 24 V nebo 230 V resp. jiné napětí vyskytující se ve stroji,
5. ovládání alespoň čtyř silových výstupů pro připojení periferií - např: maják, siréna, nebo jiné zařízení.

Druhou částí práce bude vývoj serverové aplikace, která bude data jednotlivých modulů zpracovávat. Aplikace umožní správu vstupů i výstupů a následně jejich vyhodnocení - například: počet zapnutí stroje, počet určitých operací a podobně.

### DOPORUČENÁ LITERATURA:

[1] IBRAHIM, Dogan a Ahmet IBRAHIM. The Official ESP32 Book. Holandsko: Elektor International Media, 2017. ISBN 978-1-907920-63-9.

[2] ALBERT, Lozano-Nieto. RFID Design Fundamentals and Applications. CRC Press, 2010. ISBN 9781420091250.

**Termín zadání:** 3.2.2020

**Termín odevzdání:** 1.6.2020

**Vedoucí práce:** doc. Ing. Petr Číka, Ph.D.

**Konzultant:** Jan Martinec, SRS Group s.r.o.

**prof. Ing. Jiří Mišurec, CSc.**  
předseda oborové rady

### UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

## ABSTRAKT

Cieľom diplomovej práce je preštudovať možnosti vytvorenia monitorovacieho systému pre staršie výrobné zariadenia už v jestvujúcej prevádzke. Vybrať voči charakteru zariadení potrebné komponenty pre vytvorenie systému, ktorý je schopný snímať a poskytovať informácie o činnostiach na zariadení. Ďalej navrhnúť vhodný komunikačný model pre prehľadné informovanie o prevádzke, odosielanie a ukladanie údajov. Tretím cieľom je analyzovanie dvojjadrového mikroprocesoru ESP32 voči svojim predchodcom v úlohe riadiacej jednotky systému.

## KĽÚČOVÉ SLOVÁ

ESP32, monitorovací systém, výroba, výrobné zariadenia, návrh, tvorba, realizácia, MQTT, NodeRed, Altium Designer, hardvér, doska plošných spojov

## ABSTRACT

The aim of the diploma thesis is to study the possibilities of creating a monitoring system for older production facilities already in operation. Select the necessary components for the nature of the equipment to create a system that is able to capture and provide information about activities on the equipment. The second goal is to design a suitable communication model for clear information about the operation, sending and storing data. The third goal is to analyze the dual-core ESP32 microprocessor against its predecessors in the role of system control unit.

## KEYWORDS

ESP32, monitoring system, production, production equipment, design, creation, implementation, MQTT, NodeRed, Altium Designer, hardware, printed circuit board

BORSUK, Adam. *Monitoring výrobních zařízení*. Brno, 2020, 90 s. Diplomová práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedúci práce: doc. Ing. Petr Číka, Ph.D.

## VYHLÁSENIE

Vyhlasujem, že svoju diplomovú prácu na tému „Monitoring výrobních zařízení“ som vypracoval samostatne pod vedením vedúceho diplomovej práce, s využitím odbornej literatúry a ďalších informačných zdrojov, ktoré sú všetky citované v práci a uvedené v zozname literatúry na konci práce.

Ako autor uvedenej diplomovej práce ďalej vyhlasujem, že v súvislosti s vytvorením tejto diplomovej práce som neporušil autorské práva tretích osôb, najmä som nezasiahol nedovoleným spôsobom do cudzích autorských práv osobnostných a/alebo majetkových a som si plne vedomý následkov porušenia ustanovenia § 11 a nasledujúcich autorského zákona Českej republiky č. 121/2000 Sb., o práve autorskom, o právach súvisiacich s právom autorským a o zmene niektorých zákonov (autorský zákon), v znení neskorších predpisov, vrátane možných trestnoprávných dôsledkov vyplývajúcich z ustanovenia časti druhej, hlavy VI. diel 4 Trestného zákonníka Českej republiky č. 40/2009 Sb.

Brno .....

.....

podpis autora

## POĎAKOVANIE

Rád by som poďakoval vedúcemu diplomovej, ale i predchádzajúcej bakalárskej práce, pánovi doc. Ing. Petrovi Číkovi, Ph.D. za ľudský, odborný a hlavne férový prístup. A to nielen počas tvorby prác, ale i počas celého štúdia na VUT. Ďakujem tiež pánovi konzultantovi Ing. Jánovi Martinčekovi za podnetné a odborné rady v práci. Pánovi Robovi Slobodovi za poskytnutú príležitosť. Pánovi Ing. Vladimírovi Borovskému za rady a siahodlhé rozhovory. Chcel by som v neposlednom rade poďakovať mojim rodičom za psychickú podporu počas celého štúdia. Ďakujem im zato, že to so mnou vydržali, vzhľadom na moju silnú nervozitu a nestabilnú náladu pred akýmikoľvek skúškami.

# Obsah

Úvod	12
<b>1 Návrh monitorovacieho systému</b>	<b>13</b>
1.1 IoT	13
1.2 Automatizácia a monitoring	14
1.3 Analýza výberu riadiacej jednotky	16
1.3.1 ESP8266 vs ESP32	16
1.3.2 ESP32	17
1.3.3 Zbernice	20
1.3.4 Moduly ESP32	22
1.4 Periféria	24
1.4.1 Identifikácia osôb	24
1.4.2 Záloha dát a času	27
1.4.3 Displej	28
1.4.4 Snímanie napätia	30
1.4.5 Ovládanie záťaží	32
1.4.6 Blok napájania	34
1.5 Návrh blokovej schémy	38
1.6 Programovanie	39
1.6.1 Arduino IDE	39
1.6.2 Micropython	39
1.6.3 ESP-IDF	40
1.7 Komunikácia	41
1.7.1 HTTP vs MQTT	41
1.7.2 Eclipse Mosquitto Broker	45
1.7.3 Node-RED	46
1.7.4 WAMP	47
1.7.5 Komunikačná schéma	47
<b>2 Návrh dosky plošných spojov</b>	<b>48</b>
2.1 Altium Designer	48
2.2 Schéma zapojenia	49
2.2.1 Napájanie	49
2.2.2 Obvod reálneho času	52
2.2.3 SD karta	52
2.2.4 Ovládanie externých výkonov/ snímanie prítomnosti napätia	53

2.2.5	Počítadlo pracovných cyklov . . . . .	54
2.2.6	Externé periféria . . . . .	55
2.2.7	Rozloženie na DPS . . . . .	56
2.3	Programové vybavenie . . . . .	57
2.3.1	Setup() . . . . .	57
2.3.2	Body prerušenia - Interrupts . . . . .	59
2.3.3	Pripojenie k AP a MQTT brokeru . . . . .	63
2.3.4	Identifikácia zamestnanca . . . . .	64
2.3.5	Záznamy na SD kartu . . . . .	64
2.3.6	Napájanie, nabíjanie . . . . .	67
2.4	Webové rozhranie . . . . .	70
<b>Záver</b>		<b>72</b>
<b>Literatúra</b>		<b>74</b>
<b>Zoznam symbolov, veličín a skratiek</b>		<b>77</b>
<b>Zoznam príloh</b>		<b>80</b>
<b>A Schémy</b>		<b>81</b>
A.1	Schéma vstupov a výstupov MPC23017 1/2 . . . . .	81
A.2	Schéma vstupov a výstupov MPC23017 2/2 . . . . .	82
A.3	Napájanie/regulátory . . . . .	83
A.4	Aktualizovaný obvod nabíjania . . . . .	84
A.5	Obvod prepínania napájania s LTC4412 . . . . .	85
<b>B 3D náhľady</b>		<b>86</b>
B.1	Doska s 3D modelmi . . . . .	86
B.2	Doska bez 3D modelov . . . . .	87
B.3	Bočný 3D pohľad . . . . .	88
B.4	Náhľad na medenú vrstvu . . . . .	89
B.5	Pohľad do dashboardu . . . . .	90



# Zoznam obrázkov

1.1	Rozloženie pinov a ich funkcií na ESP32 [5]	18
1.2	Topológia zbernice ISP	21
1.3	Otvorené púzdro iButton	24
1.4	Modul RFID s RC522 a integrovanou anténou	26
1.5	0,96 palcový OLED displej	29
1.6	TFT displej	29
1.7	Schéma deliča	30
1.8	Optočlen s MOSFET Toshiba TLP350 [13]	30
1.9	Všeobecná schéma detekcie striedavého napätia	31
1.10	Všeobecná schéma relé ako vstup	32
1.11	ULN2803	33
1.12	Indukčný snímač LJ12A3-4-Z/BX	34
1.13	Prúdové odbery a sumár potrebného výkonu	34
1.14	MEAN WELL IRM-10-12	35
1.15	Stabilizátor LM7805(TO220)	35
1.16	Ukážková schéma zapojenia LTC4415	36
1.17	Ukážková schéma zapojenia LTC4054	37
1.18	Návrh blokovej schémy	38
1.19	Schéma tvorby aplikácie pre ESP32 s ESP-IDF[21]	40
1.20	Schéma výmeny správ	43
1.21	Logo Mosquitto	45
1.22	Overenie behu servera	45
1.23	Subscriber	45
1.24	Publisher	46
1.25	Logo Node-RED	46
1.26	Komunikačná schéma	47
2.1	Schéma primárneho zdroja	49
2.2	Stabilizátor 5 V	49
2.3	Schéma nabíjačky	50
2.4	Schéma prepínania napájania	51
2.5	Schéma RTC	52
2.6	Schéma pripojenia SD karty	52
2.7	Schéma MCP23017	53
2.8	Schéma pripojenia indukčného snímača	54
2.9	Schéma pripojenia indukčného RFID a displeju	55
2.10	Schéma pripojenia externej antény	55
2.11	Vývojový diagram štartu časť 1	57

2.12	Vývojový diagram štartu časť 2 . . . . .	58
2.13	Vývojový diagram štartu časť 3 . . . . .	58
2.14	Vývojový diagram synchronizácie času . . . . .	69
2.15	Webové/grafické rozhranie . . . . .	70
2.16	Tvorba vzťahových funkcií . . . . .	71
A.1	Prvá časť schémy vstupov a výstupov MPC23017 . . . . .	81
A.2	Druhá časť schémy vstupov a výstupov MPC23017 . . . . .	82
A.3	Vstup napájania a regulátory napätia . . . . .	83
A.4	Schéma obvodu nabíjania a detekcie stavov nabíjačky . . . . .	84
A.5	Schéma obvodu prepínania napájania . . . . .	85
B.1	Pohľad na 3D verziu so zapnutými modelmi . . . . .	86
B.2	Pohľad na 3D verziu s vypnutými modelmi . . . . .	87
B.3	Bočný 3D pohľad so zapnutými modelmi . . . . .	88
B.4	V 3D zvýraznenie spojov . . . . .	89
B.5	Aktívny dashboard . . . . .	90

# Zoznam tabuliek

1.1	Porovnanie parametrov [3][4] . . . . .	17
1.2	UART pinout . . . . .	20
1.3	SPI pinout . . . . .	22
1.4	Verzie modulov ESP32 . . . . .	22

## Zoznam výpisov

# Úvod

V dnešnej dobe, kedy sa do popredia dostávajú nové generácie sietí, navyšuje sa prenosová rýchlosť, do sietí sa pripája čo raz viac nových zariadení a nastupuje obdobie IoT, prichádza otázka čo so staršími zariadeniami, ktoré nedisponujú novými komunikačnými technológiami. Tak ako u nových zariadení, či už pre osobné použitie alebo vo výrobných halách, existuje potreba mať všetko a vždy pod kontrolou, mať všetko neustále „online“.

Nie je možné vo výrobe nahradiť všetky staršie stroje, ktoré stále slúžia a generujú kapitál, nahradiť novými strojmi s novodobou komunikačnou výbavou, takáto výmena by stála nemalé finančné prostriedky. Preto je potrebné vytvoriť zariadenie, ktoré bude doplnkom pre staršie zariadenie a bude poskytovať informácie, ktoré sú potrebné tak aby sa dala navyšovať pracovná efektivita i týchto strojov. Či už ide o hardware implementáciu ako samotný modul v podobe DPS, alebo o internú software výbavu modulu až po komunikačnú časť a následné ukladanie a zhodnotenie vygenerovaných dát.

Pre tieto účely je potrebné aby bol modul osadený napájacou časťou so základným napájaním, výpočtovou a komunikačnou jednotkou, rozhraniami pre snímanie a kontrolu napätí, rozhrania pre ovládanie externých výkonových zariadení, snímanie počtu cyklov, ktoré stroj vykonal ale i kto na zariadení pracuje, resp. identifikáciu zamestnanca, zobrazovaciu jednotku v podobe displeja.

Na druhej strane je potreba všetky vygenerované dáta odosielať, zobrazovať a ukladať, preto je potrebné vytvoriť komunikačnú schému pomocou, ktorej sa budú dáta dostávať ku koncovému spotrebiteľovi. Výber vhodných rozhraní, protokolov ale i software výbavy až po koncové zobrazenie napríklad. pomocou webového prehliadača. S celkovou tvorbou prichádza potreba výberu programovacieho prostredia v ktorom sa bude tvoriť software pre prácu samotného modulu.

Práca je preto rozdelená na celky ako návrh a výber prvkov, ktoré tvoria hardware, a následne návrh a výber všetkých potrebných programov a aplikácií pre software časť.

# 1 Návrh monitorovacieho systému

Prvá kapitola sa zaoberá dôvodom potreby vzniku samotného monitorovacieho zariadenia, návrhom a výberom vhodných komponentov, návrhu blokovej schémy, nárokom na spotrebu energie, návrhu potrebných samostatných schém daných blokov.

## 1.1 IoT

Internet vecí (Internet of Things, IoT) je v posledných rokoch rýchlo sa rozvíjajúci trend v oblasti komunikácie a kontroly vecí (zariadení) či už medzi sebou navzájom alebo s človekom. V tomto odvetví sa najčastejšie využívajú bezdrôtové prenosy dát či už ide o menšie siete založené na štandarde 802.11 Wifi, poprípade 802.15 Bluetooth, ale i väčšie siete ako napríklad dnes poskytovaná sieť 4. generácie LTE resp. EPS, a prichádzajúce siete 5. generácie.

Pripojené zariadenia umožňujú zber množstva dát, ktoré sa ďalej spracovávajú, vyhodnocujú, vytvárajú interakcie medzi sebou a podobne. Nasadenie si nachádza priestor v takmer všetkých odvetviach priemyslu a obchodu ako je logistika, energetika a elektrotechnika, inteligentné domy, zdravotníctvo, ťažký priemysel, výrobné závody, doprava, bankovníctvo, meteorológia atď.

S tým súvisí i termín Priemysel 4.0, ktorý predpokladá zjednotenie komunikačných mechanizmov, plnohodnotnej digitalizácie a plnej automatizácie výroby. Najviac je však zameraný nato aby sa najčastejšie jednoduché úkony prestali vykonávať ľuďmi a prevzali tieto činnosti autonómne systémy.

S narastajúcim počtom zariadení stúpa i náročnosť na dátové úložiská čo dáva možnosť vzniku rôznym cloudovým službám, potreba autonómnosti zase zase vytvára nárok na tvorbu čoraz lepších algoritmov strojového učenia a umelej inteligencie. IoT vyžaduje škálovateľný sieťový priestor, preto i protokol IPv6 hrá veľkú rolu v sieťovej vrstve, ale i príchod jednoduchších aplikačných protokolov ako MQTT, CoAP a podobne. Zároveň je potreba zabrániť nadrozmernému toku dát cez internet a preto sa nasadzuje tzv. výpočet v hmle (fogging), teda výpočtové doplnenie cloudu.

Súvisiacim aspektom je i bezpečnosť a súkromie, inteligentné systémy pripojené do internetu sú vystavené zvýšenému riziku zraniteľnosti. IoT zariadení sa problematika dotýka rovnako ako serverov, počítačov, smart telefónov. Nakoľko IoT sa týka hlavne zariadení s malým vlastným výpočtovým výkonom, musia ich chrániť vyššie štruktúry ako firewall, antivírusy, antimalware ale i bezpečnostné záplaty a aktualizácie, používanie zabezpečených verzii aplikačných protokolov, dnes už ale i menšie výpočetné jednotky obsahujú hardware implementáciu kryptografických metód.

Ďalšou súčasťou je i že systém môže síce prispieť ku komunikačnému rozvoju do rôznych oblastí života, ale na druhej strane ide i vedie k ohrozeniu súkromia jednotlivca, úniky citlivých informácií firmám, sledovanie, sociálna manipulácia. Obrovské objemy dát v cloudoch poskytujú rôznym inštitúciám sa dostať k týmto dátam a pokusy o ich speňaženie (vydieranie, predaj dát tretím stranám, predaj dát konkurečnej firme a pod.), na toto riziko v posledných rokoch upozorňuje hlavne organizácia ACLU (American Civil Liberties Union). [1]

## 1.2 Automatizácia a monitoring

Prvé zmienky o zariadeniach, využívajúcich automatické prvky, siahajú až do antického sveta. V období približne 50 rokov pred naším letopočtom bol opísaný spôsob regulácie výšky hladiny vodných tokov. Herón z Alexandrie, ktorý približne v rokoch 50 až 70 nášho letopočtu zostrojil viacero automatických zariadení, napríklad veterný mlyn alebo predchodkyňu dnešnej parnej turbíny. Masívnejší rozvoj automatizácie nadišiel v 16. a 17. storočí, kedy mnohí vynálezcovia konštruovali zariadenia na automatizáciu rôznej ľudskej činnosti. Vznikali regulátory pre veterné a vodné mlyny. Jedným z významných objavov tejto doby bol Wattov odstredivý regulátor, ktorý sa používa v niektorých zariadeniach dodnes. Wattov odstredivý regulátor pôvodne slúžil na reguláciu otáčok parného stroja. Funguje tak, že regulátor sa roztáča otáčavým pohybom regulovaného zariadenia. K automatizácii výrobných procesov významne prispel Henry Ford. Ten v roku 1913 ako prvý na svete začal vo svojej fabrike pásovú výrobu automobilov. A tak ako sa vyvíjala automatizácia, začal popri tom vznikať i systém monitorovania hodnôt z týchto systémov. A tak automatizácia a monitoring sa stali navzájom neoddeliteľnými prvkami, ktoré dnes spoločne využívajú pokroky v strojárstve, mechanizácii, elektrotechnike a komunikačných technológiách.[2]

Dôvody vzniku monitorovacieho systému vyplývajú z narastajúcich požiadaviek samotnej výroby, čo súvisí najmä z navýšením počtov vykonávaných úkonov na zariadeniach, navýšenie počtu zákaziek, a tým súvisiaceho narastajúceho počtu zamestnancov. Preto je potrebné zaviesť kontrolu i starších zariadení, ktoré nedisponujú internými kontrolnými systémami s príslušnou interpretáciou. Pri zvýšení počtu úkonov, ktoré musia zariadenia vykonať sa očakáva samozrejme nárast poruchových stavov, ktoré musia byť včas identifikované a vyriešené a tým zefektívniť využitie daného stroja a celého výrobného procesu a optimálnej tvorby výrobných nákladov.

Preto je potrebné do systému zaviesť nielen samotnú kontrolu zariadenia, ale i ďalšie premenné ako:

- kto prácu na zariadení vykonáva
- akú zákazku vykonáva
- dátum, čas a dobu práce
- počet vykonaných úkonov
- poruchy, prestoje, servis
- prítomnosť všetkých potrebných napájacích vetví
- signalizácia poruchových stavov
- a pod.

To priamo súvisí s výberom vhodných modulov, obvodov a funkčných prvkov, ktoré je potrebné implementovať do systému aby boli tieto požiadavky zabezpečené a zároveň výber vhodnej komunikačnej jednotky pre zobrazovanie aktuálneho stavu a uloženie potrebných dát pre spätnú kontrolu stavu výroby. Zároveň je potrebné získané dáta rýchlo a prehľadne zobraziť a zálohovať. Treba tiež uvažovať nad množstvom a kvalite danej informácie, napríklad či potrebujem napätia presne merať alebo stačí snímať ich prítomnosť, aké veľké záťaže sa budú spínať a tým vhodný výber. Ako čo najjednoduchšie identifikovať zamestnanca, koľko zamestnanec vyrobil, aký čas nad tým strávil, bolo zariadenie viac v poruche alebo v prevádzke. To všetko ovplyvňuje čo všetko sa musí v systéme vyskytovať.



## 1.3 Analýza výberu riadiacej jednotky

V systéme je potrebné riadiť väčšie množstvo vstupných a výstupných periférií a zároveň je kladená podmienka schopnosti komunikovať s databázou cez bezdrôtovú komunikáciu Wifi. Pripojené zariadenia pracujú na rôznych zberniciach a preto je potrebné zabezpečiť modul, ktorý danými zbernicami disponuje. Zároveň je potrebný dostatočný výpočtový výkon pre zvládnutie všetkých úkonov. V tomto bode sa ponúka výber medzi dvomi zariadeniami ESP8266 a ESP32.

### 1.3.1 ESP8266 vs ESP32

Oba čipy pochádzajú od firmy Espressif Systems, pôvodný čip ESP8266 vznikol ako doplnok pre bezdrôtovú komunikáciu pre Arduino, no nakoniec vďaka jeho výpočtovej výkonnosti a neskoršieho sprístupnenia GPIO pinov pre programátorov sa stal samostatne pracujúcim modulom. ESP8266 disponuje jedno-jadrovým 32 bitovým procesorom Xtensa LX 106 s inštrukčnou sadou RISC so základným taktom 80 MHz s pretaktovaním.

Veľkosť RAM v základnej verzii je len 96 kB a takmer polovicu zaberá len Wi-Fi, ktorá zároveň beží v takzvanom „blobe“ (binárny balík bez zdrojových kódov), takže nie je optimalizovateľná. Dostupný zbytok RAM sa preto vyčerpá veľmi rýchlo. Nakoľko Wifi komunikácia zdieľa s výpočtovou časťou spoločné prostriedky a to často vedie k nestabilite buď pri prenose dát alebo výpočtovo náročnejších procesoch, je neustále potrebné vracieť programové riadenie Wifi časti a tak klasické programovanie systémom setup/loop od Arduina tu celkom nefunguje správne. Zároveň obsahuje len jeden vstup pre analógovo digitálny prevod, taktiež disponuje len slabším zabezpečením WEP.

Tento problém vyriešil nástupca - čip ESP32, ktorý disponuje dvojjadrovým procesorom Xtensa dual-core 32-bit LX6 so základným taktom 160 MHz, ktorý oddeľuje wifi časť od výpočtovej. Tento čip zároveň disponuje väčším počtom GPIO, väčším počtom zberníc, Bluetooth rozhraním ako je možné vidieť v tabuľke 1.1, obsahuje taktiež dva ADC prevodníky typu SAR s 12 bitovým rozlíšením a 18. kanálmi. V základe disponuje 400 kB RAM, pribudli i DAC prevodníky, kapacitné snímače dotyku.

Tab. 1.1: Porovnanie parametrov [3][4]

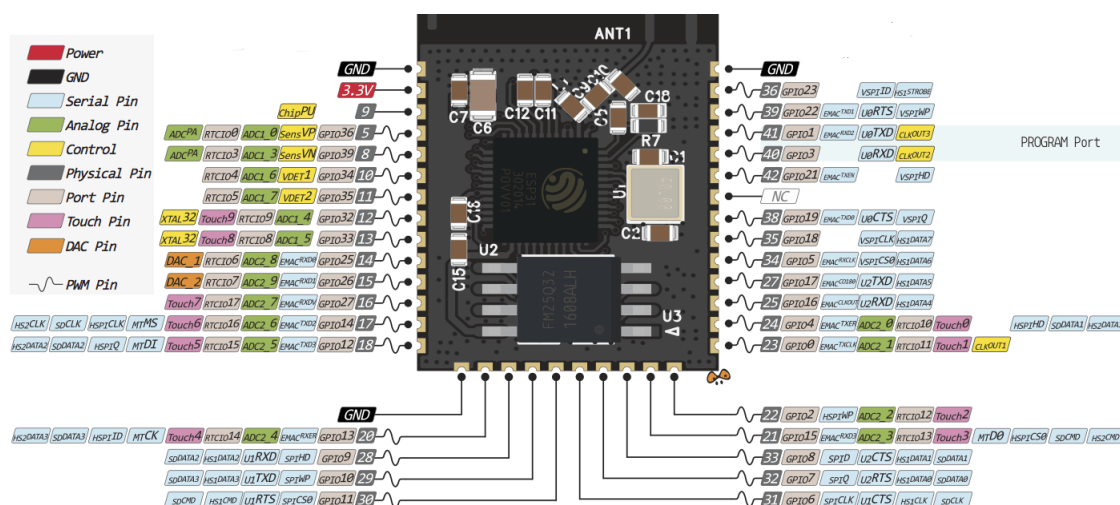
Platforma	ESP8266	ESP32
Procesor	Xtensa LX106	Xtensa dual-core LX6
Takt	80 (160) MHz	160 (240) MHz
SRAM	160 kB	512 kB
FLASH	podľa dosky	podľa dosky
Napájanie	5 V	5 V
Prac. napätie	3,3 V	3,3 V
Prúd na I/O pin	20 – 40 mA	20–40 mA
Wifi	802.11 b/g/n HT20 (max. 130 Mbit/s)	802.11 b/g/n HT40 (max. 300 Mbit/s)
Zabezpečenie	WEP	WEP, WPA, WPA2
Wifi mód	STA/AP/Mixed	STA/AP/Mixed
SPI/I2C/I2S/UART	2/1/2/2	4/2/2/2
Bluetooth	nie	v.4.2/BLE
ADC	10 bit	12 bit
GPIO	17	36
Soft PWM	8	16

### 1.3.2 ESP32

V tomto projekte je pre lepšie parametre vybraný čip ESP32. Ako je spomenuté vyššie, čip je založený na dvojjadrovom procesore so základným taktom 160 MHz s možnosťou pretaktovania na 240 MHz, oproti ESP8266 disponuje hardvérovou implementáciou MAC vrstvy a celého IP stack. Zväčšená bola veľkosť SRAM pre ukladanie premenných a výpočtov počas behu programu, veľkosť pamäte Flash a výskyt PSRAM závisí od použitej verzie modulu.

Implementované boli i interné časovače - watchdog, RTC a 64 bitový časovač a každý z nich je delený do ďalších subčasovačov s rôznymi úlohami. Prišlo taktiež k vylepšeniu Wi-Fi časti, ktorá podporuje 40 MHz pásmo spolu s modernejšími zabezpečeniami WPA a WPA2 a maximálnym vysielačím výkonom -20 dBm. Obsahuje taktiež navyše oproti ESP8266 interný teplomer, hallov senzor a pribudla i zbernica CAN Bus v2.0. Čip využíva systém vysokej integrácie a aj keď obsahuje 36 programovo dostupných pinov, tak každý z nich poskytuje viacero funkcií, ktoré sú znázornené na obr.1.1. , preto je dôležité pri návrhu jasne stanoviť využitie zberníc, využitie prevodníkov, poprípade PWM regulácie tak aby sme bola zaistená schopnosť pripojiť iné periférie na digitálne vstupy či výstupy v dostatočnom množstve.

Obrázok 1.1 graficky znázorňuje rozloženie a funkcie všetkých pinov.



Obr. 1.1: Rozloženie pinov a ich funkcií na ESP32 [5]

Nasledujúce zoznamy podrobnejšie špecifikujú funkcie daných pinov [5]:

### Input only pins

Môžu byť použité len ako vstupné piny, neobsahujú interné pull up a pull down rezistory.

- GPIO 34, 35, 36, 39

### Kapacitné snímače

ESP32 má 10 interných kapacitných snímačov, môžu snímať všetko čo vytvára a drží elektrický náboj. Je možné ich použiť ako náhrada za mechanické tlačidlá, zároveň ako prerušovacie vstupy pre prebudenie z režimu spánku.

- T0 (GPIO 4)
- T1 (GPIO 0)
- T2 (GPIO 2)
- T3 (GPIO 15)
- T4 (GPIO 13)
- T5 (GPIO 12)
- T6 (GPIO 14)
- T7 (GPIO 27)
- T8 (GPIO 33)
- T9 (GPIO 32)

## AD prevodníky

ESP obsahuje dva SAR prevodníky ADC1 a ADC2, kde prvý obsahuje 8 kanálov a druhý 10 kanálov. Každý kanál poskytuje AD prevod s 12 bitovým rozlíšením v základe so vstupným rozsahom 0–1 V s možnosťou nastavenia na 0–1,34 V, 0–2 V a 0–3,6 V. Výrobca však upozorňuje, že piny pre prevodník číslo 2 sa zároveň používajú pri bezdrôtovom prenose a preto sa počas prenosu musí riadenie týchto pinov odovzdať Wi-Fi časti.

- ADC1: CH0–CH7 (GPIO 36, 37, 38, 39, 32, 33, 34, 35)
- ADC2: CH0–CH9 (GPIO 4, 0, 2, 15, 13, 12, 14, 27, 25, 26)

## DA prevodníky

DA prevod zabezpečujú len dva kanály s rozlíšením 8 bitov.

- DAC1 (GPIO25)
- DAC2 (GPIO26)

## RTC GPIO

Tieto piny su schopné sa premapovať do nízko energetického subsystému a sú funkčné i v režime hlbokého spánku. Poskytujú možnosť pripojenia externých zdrojov prerušenia.

- |                  |                   |                   |
|------------------|-------------------|-------------------|
| • RTC 0 (GPIO36) | • RTC 8 (GPIO33)  | • RTC 14 (GPIO13) |
| • RTC 3 (GPIO39) | • RTC 9 (GPIO32)  | • RTC 15 (GPIO12) |
| • RTC 4 (GPIO34) | • RTC 10 (GPIO4)  | • RTC 16 (GPIO14) |
| • RTC 5 (GPIO35) | • RTC 11 (GPIO0)  | • RTC 17 (GPIO27) |
| • RTC 6 (GPIO25) | • RTC 12 (GPIO2)  |                   |
| • RTC 7 (GPIO26) | • RTC 13 (GPIO15) |                   |

## Strapovacie piny

Používajú sa na uvedenie ESP32 do režimu nahrávania nového programu (bootload) alebo do flash módu, kedy je nahraný kód zavádzaný z Flash pamäte.

- |          |                       |
|----------|-----------------------|
| • GPIO 0 | • GPIO 5 (boot=HIGH)  |
| • GPIO 2 | • GPIO 12 (boot=LOW)  |
| • GPIO 4 | • GPIO 15 (boot=HIGH) |

Pri týchto pinoch vzniká problém pri zapojení periférii na dané piny a pri uvádzaní ESP32 do daného módu sa stavy nemusia zhodovať a príde ku vzniku chýb.

### 1.3.3 Zbernice

ESP32 tak ako iné microprocesory disponuje niekoľkými typmi zberníc pre komunikáciu s perifériami.

#### I<sup>2</sup>C

Inter-Integrated Circuit - protokol a zbernicu vyvinula firma Philips Semiconductor. Špecifikácia zbernice I<sup>2</sup>C je založená na jednoduchých hardvérových štandardoch. Zariadenia sa delia do dvoch typov **master** - začína a končí dátový prenos generovaním štart bitu a stop bitu, generuje hodinový signál, vysiela adresu podriadeného, **slave** - odpovedá iba v prípade, že rozpoznal svoju adresu, časovanie prenosu je riadené hodinovým signálom. Zbernica používa dva vodiče - Serial Data (SDA) a Serial Clock (SCL). Na ESP32 sú defaultne nastavené GPIO 21 ako **SDA**, GPIO 22 ako **SCL**. [6]

#### UART

Universal Asynchronous Receiver and Transmitter/Univerzálny asynchrónny prijímač a vysielač je rozhranie pre sériovú komunikáciu, konvertuje dáta medzi paralelným a sériovým rozhraním. Je používaný v mikroprocesoroch pre komunikáciu s periférnym zariadením, nepoužíva hodinový signál odoslaného od vysielača k prijímaču. Dáta sa vysielaajú z pinu TX (Transmitter) a prijíma sa na pine RX (Receiver). Kľudová úroveň je log.1. Vysielať sa začne zmenou signálu na log 0 po dobu jedného bitu tzv. štart bit. Následne sa postupne z posuvného registra odosiela bit po bite od LSB po MSB, nasleduje paritný bit pre jednoduchú kontrolu (ale len ak bol predtým dohodnutý) nasledovaný stop bitom, ktorý má úroveň log.1. [6]

ESP32 disponuje tromi UART rozhraniami **UART0**, **UART1** a **UART2** (U0UXD, U1UXD and U2UXD) s pracovnou úrovňou 3,3 V TTL. UART0 je defaultne nastavený ako rozhranie pre programovanie.

Tab. 1.2: UART pinout

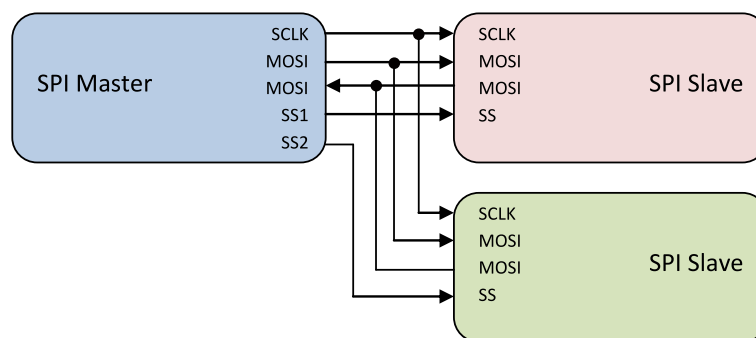
UART	RX	TX	CTS	RTS
UART0	GPIO 3	GPIO 1	N/A	N/A
UART1	GPIO 9	GPIO 10	GPIO 6	GPIO 11
UART2	GPIO 16	GPIO 17	GPIO 8	GPIO 7

## SPI

Serial peripheral interface je štvorvodičová synchronná sériová full duplex zbernica slúžiaca na prepojenie periférií s mikropočítačmi. Protokol bol definovaný firmou Motorola. Zariadenia komunikujú spôsobom master/slave, kde zariadenie master iniciuje dátový rámec. Zbernica SPI teda definuje 4 piny pre svoje použitie a na nich špecifikuje štyri logické signály

- **SCLK** – Serial Clock, časovač pre zbernicu
- **MOSI** – Master Output, Slave Input (výstup od master)
- **MISO** – Master Input, Slave Output (výstup od slave)
- **SS** – Slave Select (active low - logickú jednotku reprezentuje nižšia úroveň napätia; výstup od master)

Zbernica SPI vie pracovať s jedným zariadením master a jedným alebo viacerými zariadeniami slave vid obr. 1.2. Niektoré zariadenia vyžadujú na pine SS hranu prechodu napätia z vyššej úrovne na nižšiu pre iniciáciu akcie. Ak je pripojených viacero zariadení slave, je potrebný samostatný SS signál zo zariadenia master do každého zariadenia slave. Väčšina zariadení slave má trojstavový výstup, takže ich signál MISO sa stane hi-Z (vysoko-impedančný a neovplyvňuje obvod, do ktorého je zapojený), keď zariadenie nie je vybraté. Zariadenia bez trojstavového výstupu nemôžu zdieľať segmenty SPI zbernice s ostatnými zariadeniami. Iba jedno zariadenie môže komunikovať so zariadením master a iba jeho SS môže byť aktivované.



Obr. 1.2: Topológia zbernice ISP

Pred začatím komunikácie master najskôr nastaví hodinový signál SCLK na hodnotu frekvencie menšej alebo rovnjej maximálnej frekvencie, ktorú slave podporuje. Master potom vyberie požadovaný slave nastavením jeho signálu SS na úroveň logickej jednotky. Ak je nutná čakacia doba (napríklad pri AD prevode), potom master musí čakať minimálne po tento čas, kým začne vykonávať hodinové cykly. [6]

Počas každého hodinového cyklu sa vykoná prenos dát: master pošle bit po linke MOSI, slave ho z tejto linky prečíta, slave pošle bit po linke MISO, master ho z tejto linky prečíta.

ESP32 obsahuje celkovo 4 zbernice SPI. Avšak rozhrania **SPI0** a **SPI1** nie sú dostupné užívateľom, používajú sa interne na prístup k pripojenej Flash pamäti pomocou jednej signálovej zbernice a o ich prístupe rozhoduje interný arbiter. **SPI2** a **SPI3**, tiež označované ako HSPI a VSPI sú dve užívateľsky použiteľné, navzájom nezávislé zbernice, každá z nich obsahuje tri linky CS, takže ku každej je možné pripojiť tri podriadené jednotky SPI.

Tab. 1.3: SPI pinout

SPI	CS0*	SCLK	MISO	MOSI
SPI2/HSPI	GPIO 15	GPIO 14	GPIO 12	GPIO 13
SPI3/VSPI	GPIO 5	GPIO 18	GPIO 19	GPIO 23

Prvé pripojené zariadenie musí využiť pin CS0, následne ďalšie dve linky danej zbernice sú programovateľné.

### 1.3.4 Moduly ESP32

Espressif vyrába niekoľko verzii dosiek a modulov ESP32, tak aby priniesli vývojárom možnosť výberu modulu s najväčším potenciálom najlepšieho uplatnenia v danom návrhu. Moduly sa líšia vo veľkosti pamäte Flash, počte jadier, možnosť pripojiť externú anténu viď. tabulka 1.4 .

Tab. 1.4: Verzie modulov ESP32

ESP32 -	Chip	Flash[MB]	PSRAM[MB]	Ant.
WROOM-32	D0WDQ6	4	–	MIFA
WROOM-32D	D0WD	4/8/16	–	MIFA
WROOM-32U	D0WD	4/8/16	–	U.FL
SOLO-1	S0WD	4	–	MIFA
WROVER-PCB	D0WDQ6	4	8	MIFA
WROVER-IPEX	D0WDQ6	4	8	U.FL
WROVER-B	D0WD	4/8/16	8	MIFA
WROVER-IB	D0WD	4/8/16	8	U.FL

Vo verzii čipu prvé písmeno značí počet jadier napr.: **D**OWD vs. **S**OWD znamená **D** ako double, **S** ako solo. Ako jediná obsahuje len jedno jadro odľahčená verzia SOLO-1, ostatné disponujú plnohodnotne dvomi jadrami. Všetky moduly sú v základe dodávané so 4 MB Flash, avšak je možné si kúpiť u niektorých modulov verziu s väčšou pamäťou. Taktiež určité moduly poskytujú pripojiť externú RAM (PSRAM - pseudo static RAM) do veľkosti 8 MB a tak si navýšiť operačnú pamäť, ktorá sa plnohodnotne začlení s internou RAM. Zároveň niektoré verzie sú osadené IPEX konektorom pre pripojenie externej U.FL antény.



## 1.4 Periféria

Pre vytvorenie komplexného systému nie je potrebná len riadiaca jednotka, ale i ďalšie vstupno-výstupné periférie zabezpečujúce vykonanie úkonov vydaných riadiacou jednotkou na základe zmien v systéme.

### 1.4.1 Identifikácia osôb

Pre identifikáciu osôb resp. zamestnancov existujú dnes rôzne spôsoby identifikácie ako odtlačok prsta, iButton, či RFID identifikácia. Každý spôsob prináša výhody i nevýhody, rôzne úrovne zabezpečenia ale i úroveň pohodlia identifikácie. Keďže sa systém bude primárne integrovať do zariadení v časti kovovýroba, odtlačky prsta nepripadajú do úvahy (aj keď patria medzi najbezpečnejší spôsob) z dôvodu silného znečistenia prstov ale i drobných zranení na rukách ktoré sa v kovovýrobe vyskytujú a výrazne znižujú účinnosť takéhoto spôsobu identifikácie.

#### iBUTTON

Medzi popredných výrobcov dnes patrí hlavne spoločnosť Maxim Integrated, ktorá vlastní i ochrannú známku. Vyznačuje sa veľmi dobrou mechanickou odolnosťou, kremíkový čip na ktorom sa nachádza jedinečný 64-bitový identifikátor je uložený v ocelovom púzde o výške 16 mm. Celé púzdro tvoria dve časti, dátový kontakt tvorí viečko (lid) a uzemňovací kontakt tvorí spodná základňa puzdra (base). Navzájom sú oddelené polypropylénovou membránou. Čip využíva zbernicu 1-Wire, s dvomi módami – štandard s rýchlosťou 16 kbit/s a overdrive s rýchlosťou 142 kbit/s [7]. Dnes je základná verzia rozšírená o 20 rôznych radov, ktoré rozširujú základnú verziu:



Obr. 1.3: Otvorené púzdro iButton

- Address Only (základ, obsahuje 64 bitovú ROM)
- Memory (NVRAM, EEPROM)
- Real-Time Clock
- Secure (bezpečnostné zariadenia, elektronické zámky, parkovacie hodiny)
- Data Logger

## RFID

RFID (Radio Frequency IDentification) je ďalším vývojovým štádiom po čiarových kódach, kde vznik oboch technológií iniciovala firma Walmart pre značenie výrobkov predávaných v ich sieti obchodov. Dnes je delená na niekoľko typov podľa spoločných vlastností, a to podľa použitej frekvencie, bezpečnosti, vzdialenosti čítania, schopnosť zápisu.

### Podľa frekvencie

- **nízkofrekvenčné** – LF – (100 kHz – 500 kHz, typicky 125 a 135 kHz), dosah 0,2 až 0,5 m (podľa podmienok)
- **vysokofrekvenčné** – HF – (10 MHz – 50 MHz, typicky 13,56 MHz), dosah max. 1 m
- **ultrafrekvenčné** – UHF – (nejednotná frekvencia - typ. rozsah 860 až 960 MHz, Európa 868 MHz, USA a Canada 915 MHz), dosah cca 3 m
- **mikrovlnné** 2,45 GHz, dosah cca 2 m

### Podľa napájania

- **pasívne** – vysielateľ periodicky vysiela pulzy, ak sa v blízkosti objaví pasívny RFID čip, využije prijímaný signál pre nabitie a odošle odpoveď
- **aktívne** – sú drahšie nakoľko samotný čip disponuje vlastným napájaním, aktívne vysiela svoje údaje do okolia

### Bezpečnosť

V RFID technológii vzniklo niekoľko zabezpečovacích štandardov, ktoré implementujú rôzne stupne ochrany.

- **125 kHz** – po aktivácii ihneď vysiela údaje, podobnosť so starým spôsobom čítania magnetických pásov, riziko zobudenia karty i na väčšiu vzdialenosť a tým šanca na získanie údajov a ľahké replikovanie dát, riziko vytvorenia duplikátu s rovnakým sériovým číslom
- **Mifare štandard** – pôvodne vytvorená ako riešenie predaja cestovných lístkov, umožňuje obojsmernú komunikáciu

## Mifare verzie

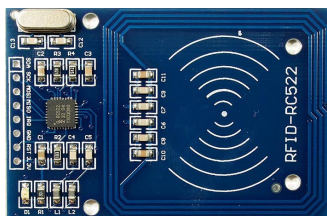
Väčšina verzií ukladá číslo karty do jednej z úložných oblastí na karte – sektory. Keď sa karta priblíži k RF polu snímača, karta a snímač začnú komunikáciu pomocou zdieľaných šifrovacích kľúčov. Po jeho zistení sa preniesie číslo karty a komunikačná relácia sa uzavrie. Tento dvojsmerný proces tiež vyžaduje viac energie ako 125 kHz, čo znamená znížený dosah čítania. Karta zároveň disponuje ďalším úložným priestorom slúžiacemu pre aplikácie do offline systémov, ukladanie kreditu a podobne [8]:

- **Mifare CSN** – karta obsahuje tzv. CSN číslo (card serial number), nie je šifrované a preto je možné ho ľahko čítať, bezpečnosťou je na úrovni 125 kHz systémov
- **Mifare Classic** – číslo je uložené v jednom zo sektorov pamäte karty, je šifrované čo sťažuje kopírovanie karty, avšak tento štandard bol prelomený
- **ICT Secured Mifare** – číslo je chránené diverzifikovaným kľúčom a šifrované pomocou algoritmu AES256
- **DESfire** – najnovší štandard, kryptografický modul je implementovaný na samotnej karte – Triple DES encryption, kryptografický modul vyžaduje viac energie, čo znižuje dosah čítania, pri priložení karty sa s ňou nesmie hýbať

Nakoľko zamestnanci disponujú kartami v štandarde Mifare, práve RFID bude začlenené do celého systému a bude poskytovať nielen informácie o danom zamestnancovi, ale i čas kedy začal úkony vykonávať, kedy skončil a koľko vyrobil. Pre správnu funkcionality RFID je potreba vybrať vhodnú verziu.

## NXP RC522

Obvod Mifare RC522 [9] od firmy NXP pracuje na frekvencii 13,56 MHz so schopnosťou čítania ale i zápisu. Podľa normy ISO/IEC 14443 A/MIFARE a NTAG podporuje verzie kariet Mifare1 S50, Mifare1 S70, UltraLight, Pro, Desfire s maximálnou prenosovou rýchlosťou 424 kBd v oboch smeroch. Je schopný pracovať na troch štandardných zberniciach SPI, UART a I2C. Maximálna vzdialenosť karty od čítačky je 6 cm. Napájacie napätie 3,3 - 5 V.



Obr. 1.4: Modul RFID s RC522 a integrovanou anténou

## 1.4.2 Záloha dát a času

Pri behu systému môže nastať niekoľko mimoriadnych situácií ako napríklad výpadok napájacieho napätia pre systém, celkový výpadok napájania i pre výrobné zariadenie, výpadok bezdrôtovej komunikácie ako zo strany ESP32 tak zo strany AP, strata spojenia s databázou. Zároveň je potrebné zapisovať časy týchto udalostí, priradovať časové známky k úkonom ako prihlásenie/odhlásenie zamestnanca, výpočet pracovného času tak aby sa o tieto dáta neprišlo a bol k nim priradený správny čas a po obnovení mohli byť tieto dáta prenesené do databázy. Samotná ESP32 nedisponuje dostatočným úložiskom ktoré by mohlo tieto štrukturované údaje uložiť v dostatočnom množstve, poskytuje síce sprístupnenie časti Flash ako SPIFFS, avšak sa dopredu u tejto pamäte počíta s častejším čítaním než zapisovaním. ESP zároveň síce disponuje interným RTC obvodom ale ten slúži skôr pre odvodenie časovania pre interné časovače. Preto je potrebné pridať väčšie úložisko ako SD karta a externý obvod reálneho času. Ku ESP32 je možné pripojiť kartu SD pomocou zbernice SPI alebo cez rozhranie SDMMC (ale len pre MMC karty).

### SD vs MMC

**MultiMedia Card**(MMC) tak ako karta SD slúži na stabilné ukladanie dát. Vytvorili ju dve firmy SanDisk a Siemens AG v roku 1997 ako odvedenie pamätí od firmy Toshiba, patrí medzi flash pamäte založené na NAND. Základný režim je 1-bitový, novšie verzie vedia pracovať v 4 a 8 bitovom režime. Dnes štandardne poskytuje kapacitu až 128 GB, vo verzii MICARD teoreticky až 2 TB. Je kompatibilná so slotom pre SD karty a pracuje na sériovej zbernici. MMC má veľkosť 24 x 32 x 1,4 mm.

**Secure Digital**(SD) patrí tiež medzi štandardné flash pamäte, vznikla v roku 1999 od spoločností SanDisk, Matsushita, a Toshiba, pretože MMC umožňovala pirátstvo dát a proti tomu sa staval hlavne hudobný priemysel.

Rozmery sú 32 x 24 x 2,1 mm, čím sa rozmerovo od MMC karty líši len hrúbkou. Vyrába sa vo viac verziách ako sú SD, miniSD a microSD. Kapacita je štandardne do 2 GB (horná hranica FAT16, ktorý nedokáže adresovať väčší pamäťový priestor). Objavujú sa i s kapacitami väčšími nevyhovujú však štandardu SD. Nástupcom je SDHC (SD High Capacity), ktorá je rozmerovo rovnaká s kapacitou do 32 GB, je však spätne nekompatibilná. Zásadný rozdiel je v adresovaní – SDHC používa adresovanie sektoru, SD adresovanie bajtu. SDHC môže použiť SD slot, nie však naopak. Využíva sériovú zbernicu. [10]

## **RTC DS3231**

DS3231 od firmy Maxim Integrated[11] je vysoko presný obvod poskytujúci hodiny reálneho času s kalendárom do roku 2100. Obsahuje kryštálový oscilátor s teplotnou kompenzáciou. RTC poskytuje informácie o sekundách, minútach, hodinách, dňoch, mesiacoch a rokoch. Dátum sa automaticky upravuje z 31 dní na nižšie podľa mesiaca, kompenzuje priestupné roky a disponuje 24 alebo 12 hodinovým formátom času. Obsahuje dva alarmy pre denné doby, programovateľný výstup a resetovací výstup. Adresa zariadenia a dáta využívajú obojsmernú komunikáciu na I2C. Obvod je schopný detekovať výpadok napájania a prepnúť na záložnú batériu. Resetovací pin môže byť využitý ako funkcia pre generovanie resetovania mikroprocesora.

- prevádzkové napätie 3,3 – 5 V
- presnosť 2 ppm pri teplotnom rozsahu 0 – 40 °C
- presnosť 3,5 ppm pri teplotnom rozsahu -40 – 85 °C
- I2C max. 400 kHz

### **1.4.3 Displej**

Priamo pri zariadení je potrebné zobrazovať základné informácie ako prihlásený zamestnanec, zákazka, prehľad stavov a podobne, čo všetko sa bude zobrazovať záleží na zhodnotení pri nasadení systému, ktoré údaje je potrebné zobrazovať. Zároveň je potrebné mať k dispozícii rozhranie pre výber a potvrdzovanie na displeji. EPS32 je schopné pracovať s displejmi cez overené zbernice ako I2C či SPI. Dnes na trhu sa ponúka veľké množstvo displejov založených na technológii LED, OLED, LCD, LCD TFT a ePaper.

#### **LCD 16x2**

Displej založený na klasickej technológii LCD s modrou farbou písma a bielym podsvietením podporujúci základné alfanumerické znaky, obsahuje dva riadky v ktorom je možné vykresliť 16 znakov, vyrábajú sa i väčšie verzie. Komunikácia pomocou I2C cez ovládač, z výroby je adresa nastavená na hodnotu 0x27 alebo 0x3F, napájacie napätie je 5 V s priemerným odberom 200 mA. Patrí medzi jednoduchšie displeje, ktorý v jednom momente nie je schopný zobrazovať viac dôležitých informácií.

## OLED I2C

Displej založený na technológii Organic LED prináša výhodu v spotrebe oproti LCD, nakoľko sa tmavé body nerozsvecujú. Veľkostne sa pohybuje od 0,96 palca po 2,4 palca, avšak vyrábajú sa i väčšie, najčastejšie vedia zobrazovať dve farby a poskytujú dostatočné rozlíšenie pre zobrazenie viacerých údajov.

**0,96 palcový displej** obsahuje jednočipový CMOS OLED ovládač SSD1306, ktorý poskytuje rozhranie I2C ale i SPI. Rozlíšenie je 128x64, označované ako „rozмеры“. Poskytuje veľmi dobrý kontrast a pozorovací uhol. Spotreba sa pohybuje na úrovni 80 mA s napájacím napätím 5 V. Ovládač SSD1306[12] má internú 1 KB GDDRAM pre obrazovku, ktorá obsahuje bitový vzor, ktorý sa má zobraziť. Pamäť je usporiadaná do 8 strán (od 0 do 7) a každá stránka obsahuje 128 stĺpcov/segmentov.



Obr. 1.5: 0,96 palcový OLED displej

## TFT LCD

TFT (Thin Film Transistor) je typ LCD, ktorý využíva tenký film s tranzistormi s aktívnou maticou. TFT displej je tvorený miliónmi tranzistorov (v závislosti od rozlíšenia), z ktorých každý ovláda jeden subpixel. Tri subpixely – RGB – červený, zelený a modrý tvoria jeden pixel. Vďaka tomu je možné vykresľovať plnofarebne. Ako podsvietenie sa používa LED zdroj a modernejšie verzie ponúkajú možnosť dotykového ovládania. Tak ako OLED sa vyrábajú v rôznych veľkostiach a rozlíšeníach pracujúce so zbernicou SPI.



Obr. 1.6: TFT displej

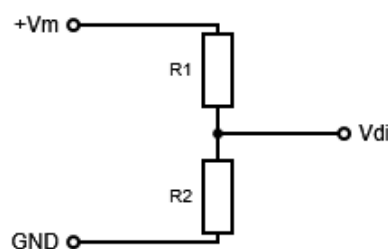
### 1.4.4 Snímanie napätia

V systéme je potrebné kontrolovať prítomnosť napájacích napätí v zariadení, ide nielen o hlavný prívod striedavého napätia 230/400 V ale i nižšie napätia v úrovniach 5 – 24 V jednosmerné i striedavé. ESP nie je schopné priamo čítať takéto napätia, vstupy sú dimenzované na 3,6 V max. s maximálnym prúdovým odberom 40 mA a síce výrobca udáva, že piny sú „5 V tolerant“ ale nie je vhodné držať stav na vstupe na piatich voltoch, taktiež nie je vhodné vždy dosahovať maximálny povolený prúd. Preto existuje niekoľko spôsobov ako to riešiť.

#### Odporový delič

Najjednoduchší spôsob ako snímať väčšie **jednosmerné** napätia je použitie klasického odporového deliča, kde si merané napätie znížime o známy pomer, tak aby sa na rezistore, z ktorého čítame hodnotu AD prevodníkom objavilo optimálne maximum 3,3 V/20 mA,

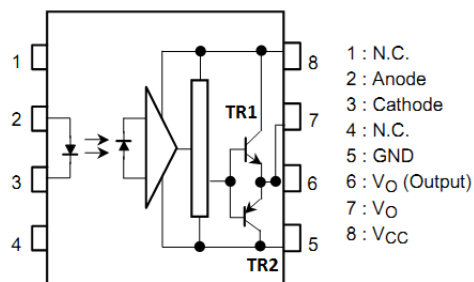
avšak objaviť sa môže i 3,6 V, čo síce vstup znesie ale AD prevodník v ESP32 na vyšších úrovniach dosahuje značnej nelinearity prevodu.



Obr. 1.7: Schéma deliča

#### Optočlen

Ďalšou možnosť ako snímať prítomnosť napätia je použiť optočlenu, ktorý prináša pri správnom výbere ochranu vstupu v kilovoltoch, pri prepätí nad schopnosť optočlena príde síce k zničeniu, ale nepríde k zničeniu vstupu, poprípade riadiacej jednotky. Je možné ho využiť v oboch smeroch, buď ako

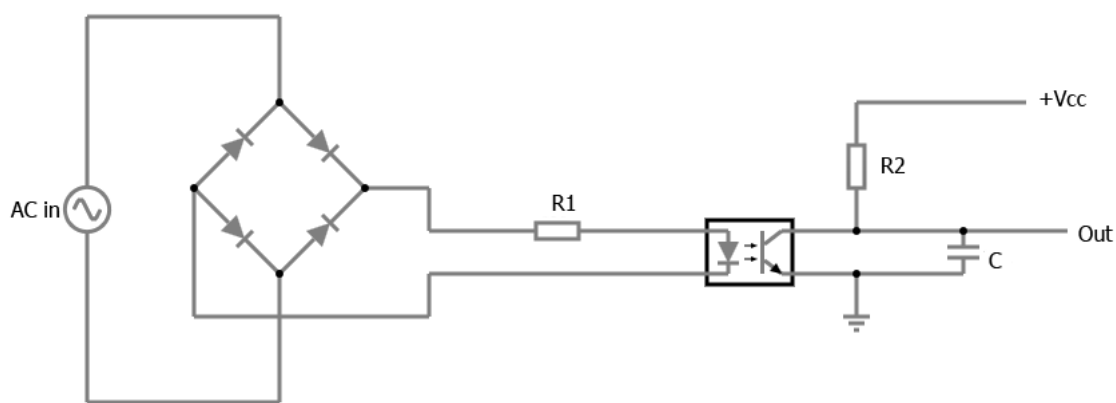


Obr. 1.8: Optočlen s MOSFET Toshiba TLP350 [13]

ochrana vstupu alebo ako výstup s ochranou pre ovládanie väčších záťaží, keďže dnes sa na trhu vyskytujú optočleny, ktoré už interne v sebe zahŕňajú NPN tranzistory, tranzistory v darlingtonovom zapojení, MOSFET/IGBT tranzistory. Tranzistory spínajú proti zemi, preto je potrebné zaradiť pull up rezistor.

## Detekcia striedavého napätia

Tak ako u iných mikroprocesorov tak i u ESP32 platí, že na vstup sa môže priviesť jednosmerné kladné napätie. Preto je potrebné, pre striedavé napätia rôznych úrovní, vykonať prevod. Často používanými obvodmi je zapojenie s optočlenom, ktorý zabezpečuje galvanické oddelenie od striedavého napätia, a bráni zavlečeniu striedavých priebehov na vstup. Súčasťou takéhoto zapojenia musí byť k úprave samotného striedavého signálu na jednosmerný pomocou diódového usmerňovača. Za usmerňovač musí byť zaradený predradný odpor aby nedošlo k poškodeniu diódy v optočlene. Výstup optočlena je spínaný cez pullup rezistor a medzi výstupné piny optočlena musí byť zaradený vhodný elektrolytický kondenzátor, ktorý eliminuje pulzáciu spôsobenú prechodom striedavého signálu cez nulu, ale ak sa kondenzátor nezaradí, je možné počítať počet prechodov/pulzov – princíp známy ako „zero crossing detection“.



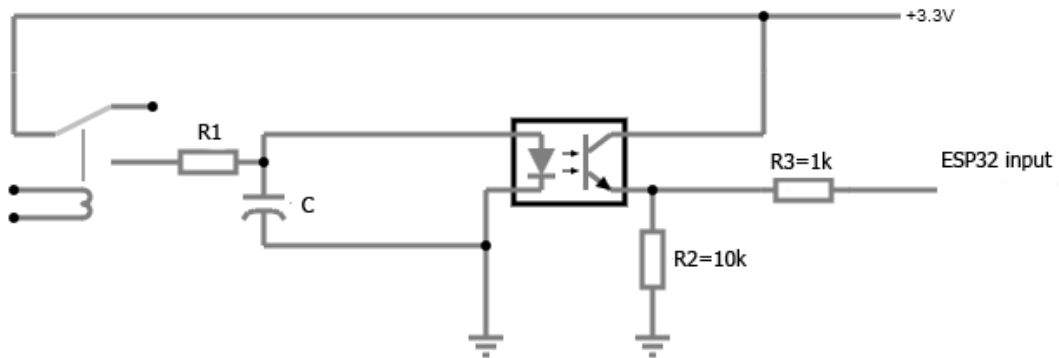
Obr. 1.9: Všeobecná schéma detekcie striedavého napätia

## Detekcia prítomnosti napätia pomocou relé

Každé relé má cievku stavanú na určité napätie (jednosmerné či striedavé), a ak sa otočí zmysel celého relé a na cievku budeme privádzať napätie, ktoré chceme detekovať je možné vytvoriť univerzálny systém detekcie prítomnosti napájania (nie merania). Kontakty NC, NO a COM sa voči GPIO správajú ako klasický vstup mechanického tlačidla, preto stačí tento vstup ošetriť pomocou pullup rezistoru, poprípade kondenzátorom, ktorý vytvorí ochranu proti zákmitom mechanického spínania. Vhodné je zaradiť na tento vstup za relé i optočlen, aby sa zabránilo pri poruche relé prerazeniu nežiadúceho napätia na vstup a zároveň sa vstup galvanicky oddelil viď obr.1.10. Takto je možné detekovať napätia, ako jednosmerné tak i striedavé, v normovaných hodnotách jednoducho a univerzálne.



V schéme nie sú rezistor **R1** a elektrolytický kondenzátor **C** určené presnými hodnotami, veľkosť rezistoru závisí na použitom type optočlenu, ktorú definuje výrobca a hodnota kondenzátora následne závisí podľa rezistoru akú časovú konštantu chceme vytvoriť ako ochranu pred nechceným spínaním diódy z mechanických zá-kmitov relé. Odpory **R2** a **R3** tvoria pull down vstup.



Obr. 1.10: Všeobecná schéma relé ako vstup

### 1.4.5 Ovládanie záťaží

ESP32, tak ako už bolo viackrát spomenuté, dokáže ovládať výstupy len do napätia 3,3 V s max. 40 mA, ak je potreba ovládať väčšie záťaže ako led pásy, krokové motory, lampy, AC motory, PWM regulácia a podobne, je potrebné doplniť medzi stupeň.

#### IRF520

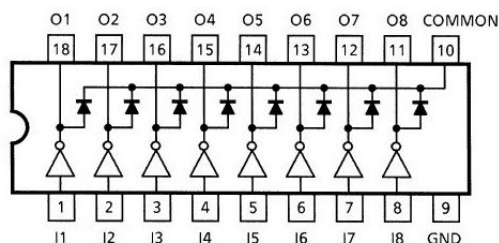
IRF520 je unipolárny N-MOSFET tranzistor, ktorý je obľúbený u Arduino nadšen-cov. Maximálne napätie je 24 V s maximálnym prúdovým odberom 2 A, avšak od jedného ampéra sa doporučuje použiť chladič. Podporuje PWM reguláciu.[6]

#### FDN306P

FDN306P je unipolárny P-MOSFET tranzistor s maximálnym napätím 12 V v smere Drain-Source a maximálnym dlhodobým zaťažením 2,6 A. Napätie v smere Gate-Source( $V_{gs}$ ) je maximálne 8 V.[23]

## ULN2803

ULN2803[14] je zariadenie obsahujúce osem NPN darlingtonových párov a na každom výstupe s ochrannou diódou pre možnosť spínať indukzívne záťaže. Maximálne dovolené napätie je 50 V a prúd 500 mA. Darlington páry môžu byť pripojené paralelne pre vyššiu kapacitu.



Obr. 1.11: ULN2803

## Relé

Tak ako pre vstupy i pre výstupy je možné použiť relé, výhodou je veľké množstvo výberu s rôznou prúdovou zaťažiteľnosťou s rôznymi úrovňami napätia. Avšak je potrebné dbať na výber relé s vhodnou cievkou, najpoužívanejšími relé s ESP32 sú s 5 voltovou cievkou, avšak pre túto ale i iné je potrebné vložiť ďalší medzistupeň pre ovládanie, buď sa použije ULN2803 alebo tak ako ponúkajú rôzne moduly je možné použiť zapojenie s optočlenom a tranzistorom, ktoré zároveň ochráni GPIO voči prúdovým rázom pri zapínaní či vypínaní cievky.

## Snímanie cyklov

V systéme je potrebné snímať počet cyklov, ktoré zariadenie vykonalo, keďže počítanie cyklov sa v týchto zariadeniach dá vykonať pomocou snímania pohybu mechanických častí je vhodné nasadiť indukčný snímač.

### Inductive proximity sensor LJ12A3-4-Z/BX (4 mm)

Indukčný snímač priblíženia LJ12A3[15] je trojvodičový, typu NPN-NO (normálne otvorený), čo znamená, že ak nie je nič zistené, je odpor čierneho vodiča vysoký. Po detekcii bude odpor nízky a pripojí sa k zemi, preto je potrebné použiť pullup rezistor. Senzor pracuje s napätím 6 – 36 V, výstupným prúdom 360 mA a pracovnou vzdialenosťou 4 mm. Je schopný detekovať kovové predmety, ale i hliníkovú fóliu.



Obr. 1.12: Indukčný snímač LJ12A3-4-Z/BX

### 1.4.6 Blok napájania

Blok napájania sa bude skladať z niekoľkých častí, pre jednoduchosť je výhodnejšie priviesť na dosku priamo 230 V a následne vytvoriť vetvy napájania pre EPS32 a periféria, nakoľko pracujú na rôznych napätiach, a tak eliminovať potrebu externých napájacích zdrojov. Pre to aby sa dalo navrhnuť napájanie je potrebné vedieť aký maximálny odber vytvoria všetky pripojené zariadenia.

Komponent	Počet	Max. odber [A]	Nap. napätie [V]	Výkon/ks [W]	Výkon spolu [W]
ESP32	1	0,65	3,3	2,145	2,145
RFID	1	0,026	3,3	0,0858	0,0858
SD Card	1	0,1	3,3	0,33	0,33
TFT Display	1	0,15	5	0,75	0,75
RTC modul	1	0,0005	5	0,0025	0,0025
Detekcia napätia	8	0,02	5	0,1	0,8
Relé	4	0,02	5	0,1	0,4
Snímanie cyklu	1	0,36	5	1,8	1,8
				Spolu	6,313

Obr. 1.13: Prúdové odbery a sumár potrebného výkonu

Keďže je potrebné vyviesť i 12 V DC linku, je vhodné použiť AC-DC menič, ktorý poskytne dostatočný výkon a napätie. Následné linky 3,3 V a 5 V sa vytvoria pomocou lineárnych stabilizátorov pevného napätia.

### Spínaný zdroj MEANWELL IRM-10-12

Spínaný zdroj do DPS, s jednosmerným výstupným napätím 12 V a prúdom max. 0,85 A (10,2 W) obsahuje ochranu proti preťaženiu a prepätiu. Vstupné striedavé napätie je 85 – 264 V a odberom 0,15 A pri 230 V a účinnosťou 0,85. Rozmery sú 45,7 x 25,4 x 21,5 mm.[16]



Obr. 1.14: MEAN WELL IRM-10-12

### Lineárne stabilizátory pevného napätia LM7805 a LM7833

Stabilizátory pevného napätia LM7805(5 V) a LM7833(3,3 V) sú polovodičové súčiastky slúžiace pre poskytnutie stabilizovaného napätia danej hodnoty pri zmenách výstupného prúdu. Typy LM patria medzi zaužívané komponenty, v ponuke sú od množstva výrobcov a poskytujú rôznu škálu maximálneho výstupného prúdu. Pre potreby projektu postačujú verzie s 1 A výstupmi.



Obr. 1.15: Stabilizátor LM7805(TO220)

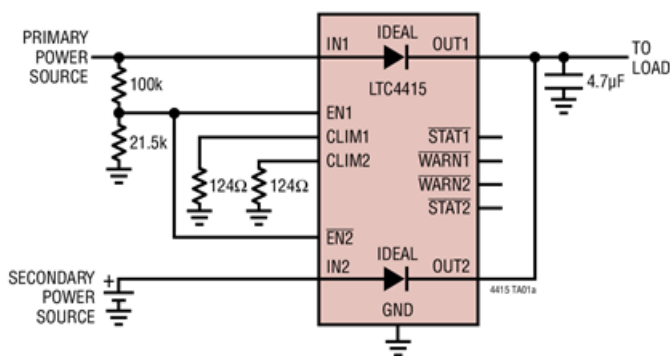
Okrem samotného primárneho zdroja napájanie, je potrebné zabezpečiť sekundárne napájanie minimálne pre EPS32 a SD kartu, aby nedochádzalo k nekorektnému stavu, čo môže spôsobiť následné chyby po reštarte ako napríklad zle zapísané stavy v EEPROM, nedokončenie prenosu, strata posledných údajov, zápis na SD kartu. ESP32 taktiež trpí chybou implementácie watchdog-u a je precitlivenejší na zámky v napájaní, poprípade pri prepínaní zdrojov. Preto je systém potrebné vybaviť sekundárnym zdrojom ako je napríklad Li-Ion batéria, ktorá bude pripojená cez obvod, ktorý bude rozhodovať o prepínaní primárneho a sekundárneho napájania s minimálnym prepínacím zákmitom, tak aby v citlivej časti výpadku neprišlo k nevyžiadanému reštartu.

## Panasonic NCR18650B 3400 mAh

Nabíjací článok, s technológiu Li-Ion, Panasonic NCR18650B[17] je priemyselná batéria poskytujúca jednosmerné napätie 3,7 V s kapacitou 3400 mAh s nabíjacou charakteristikou CC/CV a internou ochranou proti skratu. Maximálne nabíjacie parametre: 4,2 V/1,625 A. Najnižšie dovolené napätie: 2,5 V. Maximálny vybíjací prúd: 3 A podľa výrobcu, 2,5 A max. doporučené. Počet nabíjacích cyklov sa pohybuje od 500 do 1000. Tento článok poskytne dostatočnú kapacitu a napätie pre prvotné uloženie potrebných dát a korektného prechodu ESP32 do režimu hlbokého spánku, kým sa znova neobnoví primárne napájanie.

## LTC4415

LTC4415[18], od spoločnosti Linear Technology, je obvod s dvojitou ideálnou diódou s nastaviteľným prúdovým obmedzením a umožňuje spoločne prevádzkovať dva napájacie zdroje s 1  $\mu$ A záverným prúdom smerujúceho z výstupu OUT naspäť do vstupu IN. Tento obvod teda poskytuje veľmi účinné prepínanie dvoch zdrojov s ideálnymi diódami. Pracovné napätie má rozsah od 1,7 V až do 5,5 V s maximálnou zaťažiteľnosťou 4 A. V priepustnom smere je úbytok napätie len 15 mV s odporom v zopnutom stave len 50 m $\Omega$ , a tak je výrazne nižší ako pri Schottkyho diódach, zároveň tak napomáha udržať výdrž batérie a zabezpečuje, že počas prepínania nevzniknú oscilácie.

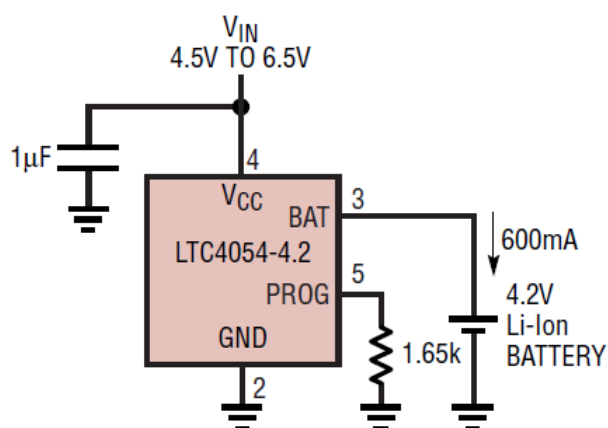


Obr. 1.16: Ukážková schéma zapojenia LTC4415

Pomocou vstupov  $EN1$  a  $\overline{EN2}$  je možné povoľovať, zakazovať alebo balansovať prioritu zdrojov pomocou ich prahových úrovní. Obvod poskytuje taktiež stavové a výstražné indikátory (nadprúdy, prekročenie teploty) a mäkký štart obmedzujúci nárazové prúdy.

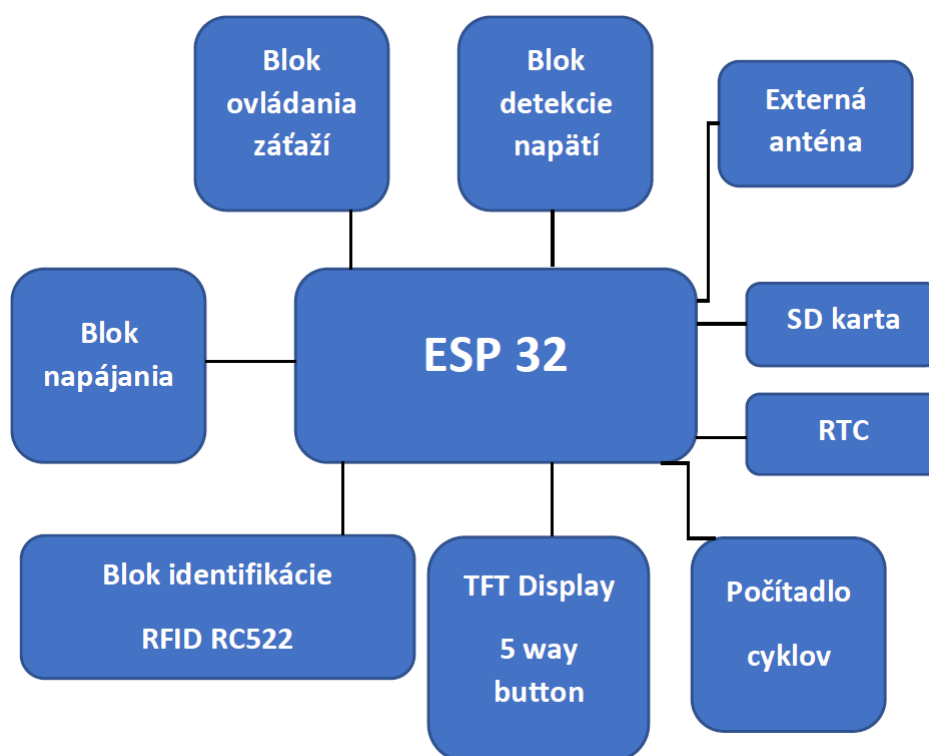
## LTC4054

LTC4054[19] je nabíjačka lítium-iónových batérií pomocou nabíjacej charakteristiky CC/CV pre jeden článok, navrhnutá tak aby podporovala napájanie z USB. Nevýžaduje externý snímací odpor a blokovaciu diódu, všetko je integrované v internej architektúre. Disponuje tepelnou spätnou väzbou pre reguláciu nabíjacieho prúdu pri náraste teploty. Nabíjacie napätie je pevné na hodnote 4,2 V a nabíjací prúd je možné programovať cez pin PROG, ktorý zároveň slúži pre monitorovanie nabíjacieho prúdu a blokáciu nabíjania, maximálny nabíjací prúd je 800 mA. Obvod automaticky ukončí nabíjanie ak hodnota prúdu klesne na jednu desatinu nastaveného nabíjacieho prúdu. Napájacie napätie je v rozsahu 4,25 V až 6,5 V, pri poklese napájacieho vstupu pod úroveň BAT príde k odpojeniu vstupu.



Obr. 1.17: Ukážková schéma zapojenia LTC4054

## 1.5 Návrh blokovej schémy



Obr. 1.18: Návrh blokovej schémy

## 1.6 Programovanie

Jednoduchý spôsob ovládania ESP32 umožňuje ovládať iba jeho Wifi jadro so základným firmvérom pomocou „AT“ príkazov cez sériovú linku. Pre ovládanie druhého jadra sa využíva napríklad firmvér NodeMCU, open-source operačný systém, ktorý umožňuje vytvorenie vlastného riadiaceho programu a ovládanie GPIO pinov. Pre NodeMCU je nutné prehrať pôvodný firmvér, ktorý je v čipe nahraný od výrobcu.

### 1.6.1 Arduino IDE

Arduino IDE je programovacia platforma založená na jazyku C a používa vlastný programovací jazyk Wiring. Vývojové prostredie pozostáva z textového editora, zobrazovania správ, konzoly a ponuky nástrojov. Pomocou IDE sa taktiež vykonáva pripojenie na hardware a nahrávanie programu. Pre vývojárov so skúsenosťami z Arduina je tento nástroj najjednoduchšia možnosť ako prejsť na výkonnejšiu platformu s podporou WiFi komunikácie. Avšak pre prácu s ESP32 je potrebné do-inštalovať časť ESP32 Core. ESP32 má v určitých častiach problém so zabehnutou štruktúrou Setup/Loop ako u Arduina. Výhodou je však veľká komunita programátorov a taktiež veľká podpora rôznych knižníc, čo podstatne uľahčuje tvorbu kódu.

### 1.6.2 Micropython

Micropython je opensource implementácia jazyka Python 3, ktorý je optimalizovaný na beh v mikrokontroléroch. Python je interpretačný jazyk, to znamená, že mikrokontrolér vykonáva príkazy napísané v Pythone bez toho, aby ich kompiloval. Pre prácu je preto potrebné doplniť interpreta pre Micropython. Na inštalovanie firmvéru do modulu ESP je určený nástroj esptool. Pri prvom nahrávaní sa odporúča vyčistiť pamäť flash pomocou funkcie erase flash. Vo Windows je potreba nástroj esptool.exe z adresy <https://dl.espressif.com/dl/esptool-2.6.1-windows.zip>.

Jednou z výhod funkcií MicroPythonu je, že poskytuje interaktívnu funkciu REPL (Read-Evaluate-Print Loop). REPL umožňuje pripojiť sa k doske a vykonáva aktuálne napísaný kód bez toho, aby sa musel neustále kompilovať a nahrávať. Nevýhodou je však menšia dostupnosť knižníc a nutnosť si určité funkcie a procedúry vytvárať samostatne.

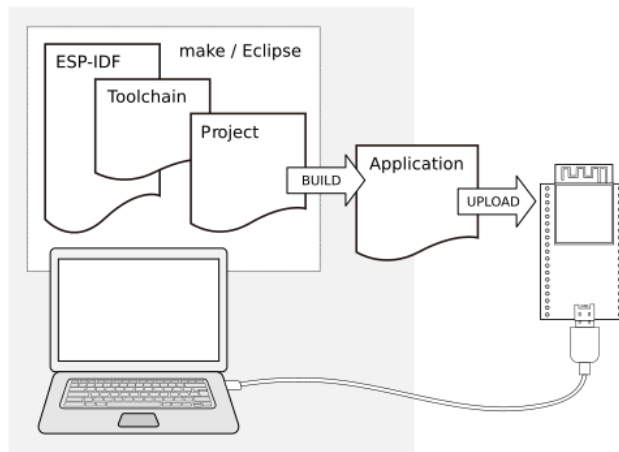


### 1.6.3 ESP-IDF

**Espressif IoT Development Framework (ESP-IDF)**[21] je framework pre tvorbu natívneho programu v ktorom sa dosahuje najvyššej kompatibility kódu voči jadru ESP32. Tvorcovia ho označujú ako „hard-core“ programovanie, keďže pri tvorbe programu sa programátor dostane viac pod kapotu fungovania jadier ESP32.

Je potrebné vytvoriť softvérový reťazec zložený z :

- Toolchain pre zostavenie kódu
- Zostavovacie nástroje CMake a Ninja pre tvorby celej aplikácie
- ESP-IDF – ktorý obsahuje knižnice a zdrojové kódy a scripty pre ovládanie Toolchainu
- Textový editor pre písanie projektov v jazyku C – napr.: Eclipse



Obr. 1.19: Schéma tvorby aplikácie pre ESP32 s ESP-IDF[21]

ESP-IDF sa doporučuje používať len ak programátor má dostatočné skúsenosti s CLI a zároveň je riadne oboznámený z ESP32 CORE určený pre Arduino IDE. Pre základnú tvorbu je výhodnejšie použiť Arduino IDE, neskôršie pre dosiahnutie vysokej kompatibility IDF.

## 1.7 Komunikácia

Okrem internej komunikácie procesora s perifériami pomocou vlastných zberníc (SPI, I2C, UART) je potrebné zabezpečiť komunikáciu pre odosielanie dát aktuálneho stavu a zároveň pre ukladanie dát do databázy cez rozhranie Wifi, ktoré bude pripojené do centrálného AP. V dnešnej dobe sa v oblasti IoT ponúkajú protokoly aplikačnej vrstvy **HTTP** (Hypertext transport protocol), **MQTT** (Message Queuing Telemetry Transport), **AMQP** (Advanced Message Queuing Protocol) a **CoAP** (Constrained Application Protocol). Avšak podstatné je to, že dnes ESP32 štandardne zvláda len protokol HTTP a MQTT, kde každý z nich prináša do komunikácie osobitné aspekty.

### 1.7.1 HTTP vs MQTT

#### HTTP

Protokol HTTP patrí medzi staršie protokoly, ktorý je avšak široko rozšírený vo verzii v.1.1 definovanej v RFC2616. Primárne sa používa na komunikáciu s WWW servermi. Je schopný prenášať formáty HTML, XML a iné. Primárne pracuje nad protokolom TCP na porte 80 a pracuje s typom komunikácie request-response (požiadavka - odozva) s modelom client-server. HTTP klient(user agent) začne požiadavku nadviazaním TCP spojenia. HTTP server počúva na danom porte, kým klient pošle reťazec s požiadavkou napr. GET, nasledovaný sériou hlavičiek opisujúcich detaily požiadavky a ľubovoľné údaje. Niektoré hlavičky sú nepovinné, zatiaľ čo verzia v.1.1 niektoré vyžaduje. Po prijatí požiadavky server pošle reťazec s odpoveďou napr. 200 OK, nasledovanou hlavičkami spolu so samotnou správou (správa, chybové hlásenie a pod.).

Druhy žiadostí :

- **GET** – žiadosť o zdroj uvedením jeho URL
- **POST** – podobné GET, okrem toho je pridané telo správy
- **PUT** – editácia údajov objektu, POST vytvára nový objekt s novými dátami, PUT upravuje dáta
- **DELETE** – zmazanie zdroja
- **HEAD** – podobné GET, prenášajú sa iba hlavičky (získanie metadát)
- **TRACE** – odošle kópiu obdržanej požiadavky späť odosielateľovi
- **OPTIONS** – vracia HTTP metódy, ktoré daný webserver podporuje
- **CONNECT** – použitie s proxy serverom, ktorý sa môže zmeniť na SSL tunel

Problémom u HTTP protokolu je však aspekt, že pri verzii v.1.0 vytvára non-perzistentné spojenie a po vrátení odpovede server okamžite ukončí spojenie a tak pri každom prenose je potreba obnovovať pripojenie. Verzia v.1.1 vytvára dočasné perzistentné spojenie a po prenose prvej správy server hneď spojenie nezavrie ale načúva či neprichádzajú ďalšie príkazy, ak do určitého časového limitu neprídu tak spoj uzavrie, dnes tento problém čiastočne rieši príkaz **Keep-Alive** a čaká až kým klient neuzavrie spojenie správou **Close**. Toto však prinieslo problémy hlavne pre tvorcov web-shopov, ktorý musia neustále udržiavať informáciu o prihlásenom zákazníkovi (dnes sa to rieši pomocou „cookies“) Nevýhodou protokolu HTTP je, že je viac „ukecaný“, hlavne kvôli hlavičkám a potrebnému obnovovaniu spojenia, protokol neimplementuje žiadne úrovne **QoS** a v neposlednom rade neprenáša vzájomnú komunikáciu dvoch klientov a je to potrebné riešiť samostatnou aplikáciou na úrovni servera.

## MQTT[22]

Message Queuing Telemetry Transport je jednoduchý **M2M(Machine to Machine)/IoT** komunikačný protokol. Pôvodne za protokolom stoja dvaja programátori Dr.Andy Stanford-Clark z IBM a Arlen Nipper z Arcom(dnes Eurotech) už v roku 1999. Dnes protokol spravuje konzorcium Eclipse foundation a dostal štandard OASIS (Organization for the Advancement of Structured Information Standards). Pracuje nad protokolom TCP štandardne na porte **1883**, ktorý bol pre tento protokol rezervovaný od IANA a port **8883** pre MQTT cez SSL. Využíva návrhový vzor publisher-subscriber a využíva pritom jeden centrálny bod (MQTT broker – „novinár“), ktorý sa stará o vzájomnú výmenu správ.

## Správy

Správy sú delené na témy(topic) a zariadenie môže publikovať v danej téme (publisher), posiela dáta na broker, alebo môže prijímať tému o ktorú sa prihlási(subscriber). Výhodou je, že jedno zariadenie môže byť v jednej téme publisher a v inej zase subscriber. Obsah správ nie je definovaný. Broker ich považuje za binárne dáta, ktoré majú byť prenesené, avšak najčastejšie sa používa formát JSON (JavaScript Object Notation) alebo textové správy, v zásade ale môžu byť akéhokoľvek formátu do veľkosti max. 256 MB. Pre odber správ z témy nemusí poznať adresu publisher, stačí mu poznať adresu brokera a názov témy a sú prenášané len správy ku ktorým sa prihlási. Témy vytvárajú hierarchickú štruktúru oddelenú lomkami. Napríklad: snímač napätia 12 V na zariadení 2 v kovovýrobe v budove A01.

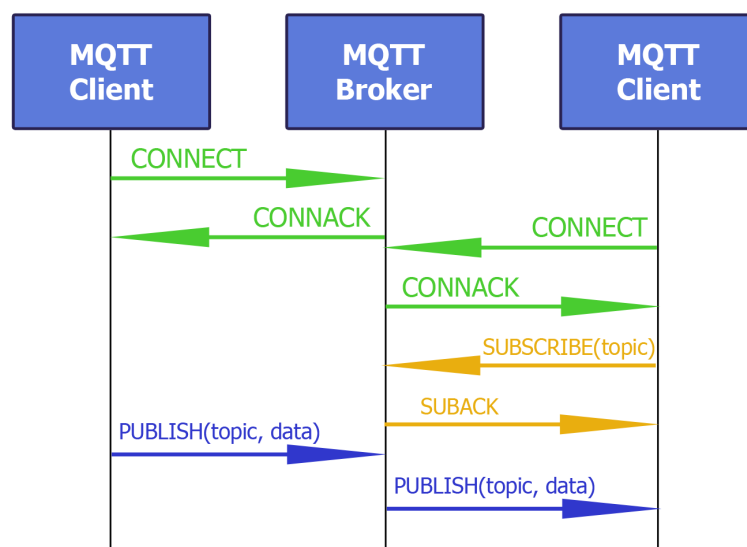
$$\text{budovaA01/KOVO/stroj2/snimac12} \quad (1.1)$$

Témy sú v podobe UTF-8 kódovania, takže sú schopné pracovať i s diakritikou, ale najlepšie je sa diakritike vyhnúť. Hierarchia tém nie je daná a všetko závisí od tvorcu ako si nastaví rozdelenie. Stáva sa, že programátorsky nastavená hierarchia nemusí korešpondovať s realitou.

### Výmena správ

Klient naväzuje spojenie s brokerom pomocou TCP na porte 1883 (pri nechránenom pripojení). Po naviazaní TCP spojenia sa posiela správa **CONNECT** s príznakom „clean session“, ktorý otvára novú komunikáciu bez príjmu tém. Broker odpovedá správou **CONNACK**, čo potvrdzuje spojenie. Nasleduje jedna alebo viac správ **SUBSCRIBE** s témami, ktoré chce zariadenie odoberať, broker odpovedá správou **SUBACK**. Pre prihlásenie k ďalším témam môže prebehnúť už kedykoľvek a taktiež môže začať posilať správy pomocou **PUBLISH**. Odhlásenie odberu danej témy sa robí správou **UNSUBSCRIBE** a broker musí odpovedať správou **UNSUBACK**. Pre úplné korektné zrušenie spojenia s brokerom sa používa správa **DISCONNECT**. Následne sa môže klient znova pripojiť so správou **CONNECT** bez príznaku a tak sa prihlási ku všetkým posledným aktívnym témam.

MQTT disponuje detekciou života zariadenia, ak zariadenie nepošle v určitom časovom rámci žiadnu správu, považuje ho za násilne odpojené a vysiela správu tzv. závet (**TESTAMENT**) a pre všetkých ostatných ho vyhlási v daných témach za odpojeného. Aby k tomuto stavu neprichádzalo zámerne, musí zariadenie, ak nemá žiadne dáta na poslanie, posilať správu **PINGREQ** a broker odpovedá **PINGACK**.



Obr. 1.20: Schéma výmeny správ

## QoS a retain

MQTT disponuje tromi úrovňami kvality služieb, naproti tomu samotný HTTP protokol nedisponuje vlastným QoS. Existuje i akoby podmultá úroveň a to bez nasadenia QoS, teda best effort. Tri úrovne v MQTT sú označené číslami 0, 1 a 2 a líšia sa v tom ako sa dbá na potvrdenie a ako silno je zaistené dodanie. Výhodou je, že nastavená kvalita sa prenáša na celom komunikačnom uzle.

V **úrovni 0** (at most once/fire and forget) pošle publisher správu PUBLISH brokeru a ďalej sa nestará.

V **úrovni 1** (at least once) zaručuje, že správa bola doručená aspoň jeden krát. Po prijatí správy brokerom ju broker ďalej odošle k aktívnym odberateľom. Po prijatí odberateľom odpovie subscriber správou PUBACK, broker správu odstráni a pošle PUBACK publisherovi. Publisher tak vie, že správa prešla cez broker a zahodí ju. Existujú dva spôsoby a to, keď broker čaká na potvrdenie od všetkých odberateľov alebo mu stačí potvrdenie od jedného, záleží od nastavenia.

Na **úrovni 2** (exactly once) zaručuje, že správa bola dodaná práve jeden krát. Opäť ako predtým prebehne odoslanie správy z publishera, broker odosiela odberateľom a zároveň posiela publisherovi správu **PUBREC**. Publisher odpovie správou **PUBREL** a broker správu zmaže a potvrdí správou **PUBCOMP** a tým sa celá výmena uzatvorí.

Pri správach sa okrem QoS prenáša i príznak „retain flag“, ktorý oznamuje brokeru, že nesmie správu po prijatí a rozoslaní zahodiť ale má ju uložiť a ponechať pre nových odberateľov. Novým odberateľom sa pošle vždy len posledná takto označená správa.

## 1.7.2 Eclipse Mosquitto Broker

Eclipse Mosquitto je broker(sprostredkovateľ správ) s open-source licenciou(EPL/EDL), ktorý pracuje s MQTT protokolovými verziami 5.0, 3.1.1 a 3.1. Výhodou tohto brokera je jeho jednoduchosť a je možné ho nasadiť na rôzne škály zariadení, od Raspberri PI cez stolné počítače až po plnohodnotné servery. Mosquitto poskytuje knižnicu C na implementáciu MQTT klientov Mosquitto\_pub a Mosquitto\_sub z príkazového riadka. Bežná implementácia spotrebuje okolo 3 MB RAM pri pripojení 1000 klientov. Výhodou Mosquitto je že okrem klasických klientov podporu aj tzv. bridge mode, ktorý umožňuje sa spojiť s iným MQTT serverom a tak môže poskytnúť zostavenie sietí serverov a tak distribuovať správy i na vzdialené pripojenia.



Obr. 1.21: Logo Mosquitto

### Overenie funkčnosti brokera

Po nainštalovaní aplikácie treba v zdrojovej zložke spustiť „mosquitto.exe“, tým sa naštartuje server(bez akýchkoľvek klientov), ktorý beží lokálne na počítači a začne počúvať na porte 1883, čo je možné overiť v príkazovom riadku vid obr. 1.22.

```
C:\Users\adam>netstat -a
```

Active Connections			
Proto	Local Address	Foreign Address	State
TCP	0.0.0.0:135	adam-PC:0	LISTENING
TCP	0.0.0.0:445	adam-PC:0	LISTENING
TCP	0.0.0.0:1883	adam-PC:0	LISTENING
TCP	0.0.0.0:2869	adam-PC:0	LISTENING

Obr. 1.22: Overenie behu servera

Následne v samostatných oknách príkazového riadka môžeme vytvoriť publishera a subscribera. V okne 1 sa vytvorí subscriber s príznakom „-d“, čo znamená že beží ako daemon a zároveň zobrazuje všetky debug správy, za príznakom „-t“ nasleduje názov témy.

```
D:\PROGRAMY\mosquito\mosquitto>mosquitto sub -d -t test
```

Okno 1

```
Client mosq-FiFBYZAp4SHeVbK3BX sending CONNECT
Client mosq-FiFBYZAp4SHeVbK3BX received CONNACK (0)
Client mosq-FiFBYZAp4SHeVbK3BX sending SUBSCRIBE (Mid: 1, Topic: test, QoS: 0, Options: 0x00)
Client mosq-FiFBYZAp4SHeVbK3BX received SUBACK
Subscribed (mid: 1): 0
Client mosq-FiFBYZAp4SHeVbK3BX sending PINGREQ
Client mosq-FiFBYZAp4SHeVbK3BX received PINGRESP
Client mosq-FiFBYZAp4SHeVbK3BX sending PINGREQ
Client mosq-FiFBYZAp4SHeVbK3BX received PINGRESP
Client mosq-FiFBYZAp4SHeVbK3BX received PUBLISH (d0, q0, r0, m0, 'test', ... (11 bytes))
Hello World
```

Obr. 1.23: Subscriber

V okne 1 na obrázku 1.23 je vidieť, že pokiaľ nebol vytvorený ešte publisher a neposielal žiadne správy, tak sa subscriber každých 60 sekúnd(default hodnota) hlásil serveru o tom, že žije ako bolo popísané i v definícii MQTT protokolu. V okne 2 na obrázku 1.24 je vidieť tvorba publisher, ktorému je pridaný príznak „-m“, teda z týmto príznakom nasleduje daná správa. Následne v okne 1 je vidieť prijatie správy s textom „Hello World“. Týmto sa overila funkčnosť samotného servera, ale i to že server korektne vytvára vlastných klientov a témy.

```

Okno 2
Microsoft Windows [Version 10.0.18362.535]
(c) 2019 Microsoft Corporation. Všetky práva vyhradené.

C:\Users\adam>D:

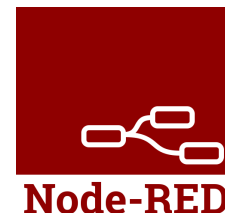
D:\>cd PROGRAMY\moskito\mosquitto

D:\PROGRAMY\moskito\mosquitto>mosquitto_pub -d -t test -m "Hello World"
Client mosq-3qAolpj2HDzkcrnGvq sending CONNECT
Client mosq-3qAolpj2HDzkcrnGvq received CONNACK (0)
Client mosq-3qAolpj2HDzkcrnGvq sending PUBLISH (d0, q0, r0, m1, 'test', ... (11 bytes))
Client mosq-3qAolpj2HDzkcrnGvq sending DISCONNECT
  
```

Obr. 1.24: Publisher

### 1.7.3 Node-RED

Node-RED je programovací nástroj na prepojenie hardvérových zariadení, API a on-line služieb. Poskytuje tvorbu JavaScriptu. Interná knižnica dovoľuje ukladať vlastné funkcie, šablóny a toky. Pracuje s formátmi JSON, XML ale i emailové správy, realizuje ich vyžiadanie a následne grafické zobrazenie, ale i odosielanie týchto dát. Poskytuje editor založený na tzv. browser-based flow, teda závislosti sa tvoria formou vývojových diagramov. Tie je možné potom nasadiť do runtime modulu. K dispozícii sú bloky odosielania a zpracovania správ, grafické ovládacie rozhranie, ktoré je dostupné cez prehliadač s odkazom na lokálnu webovú stránku.



Obr. 1.25: Logo Node-RED

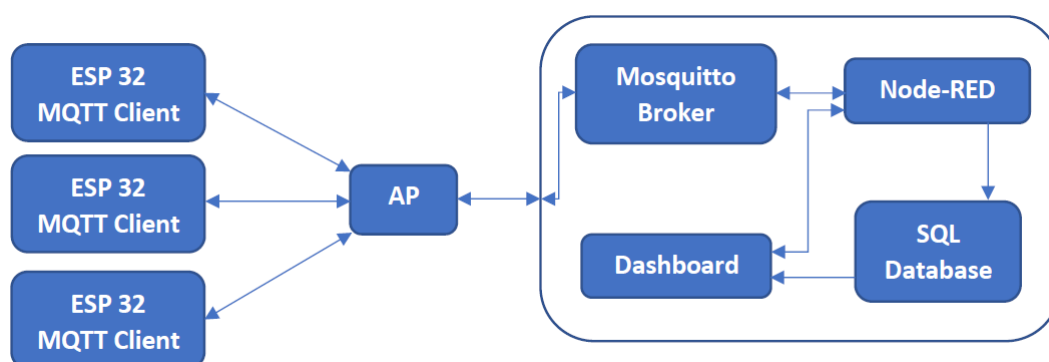
Pokiaľ do prostredia pristupujeme lokálne stačí vo webovom prehliadači zavolať stránku <http://localhost:1880> a objaví sa programovacia časť. Pre zobrazenie vytvorenej grafickej časti je potrebné zadať do prehliadača <http://localhost:1880/ui>. Je možné nastaviť aby program začal pracovať automaticky po spustení zariadenia, kde je nainštalovaný a zároveň aby vyžadoval prihlasovacie meno a heslo.

### 1.7.4 WAMP

Skratka znamená Windows, Apache, MySQL a PHP. WAMP je variáciou LAMP a XAMPP pre systémy Windows, inštaluje sa ako softvérový balík. Najdôležitejšou súčasťou balíka WAMP je Apache (Apache HTTP Server), ktorý sa používa na spustenie webového servera v systéme Windows. Spustením lokálneho webového servera Apache je možné testovať webové stránky vo webovom prehliadači bez ich zverejnenia na internete.

WAMP tiež obsahuje MySQL a PHP, ktoré sú dve z najbežnejších technológií používaných na vytváranie dynamických webových stránok. MySQL je vysokorýchlostná databáza, zatiaľ čo PHP je skriptovací jazyk, ktorý možno použiť na prístup k údajom z databázy. Inštaláciou týchto dvoch komponentov lokálne je možné vytvoriť a otestovať dynamickú webovú stránku. Aj keď Apache, MySQL a PHP sú komponenty s otvoreným zdrojom, ktoré je možné nainštalovať jednotlivo, zvyčajne sa inštalujú spoločne.

### 1.7.5 Komunikačné schéma



Obr. 1.26: Komunikačné schéma

ESP32 s nasadeným MQTT klientom sa bude hlásiť cez AP do brokera Mosquitto, Node-RED sa voči brokeru bude tiež správať ako klient a odoberať dané témy. Následne aktuálne dáta zobrazí v dashboarde v časti aktuálnych situácií, a dané dáta zároveň uloží do SQL databázy, ktorá bude uchovávať dáta pre možnosť tvorenia štatistiky a zálohy dát o stavoch daných zákaziek, stavu zariadenia a zamestnancoch pracujúcich na zariadeniach. História sa bude dať prezerať v dashboarde, to teda znamená, že bude potreba zlúčiť grafický výstup z Node-RED a SQL databázy. Implementácia Node-RED spolu s brokerom a MQTT protokol zabezpečia možnosť i posielat dáta do modulov na zariadeniach a tak vytvoriť obojsmernú komunikáciu s možnosťou čiastkového diaľkového ovládania určitých stavov na zariadeniach.



## 2 Návrh dosky plošných spojov

Táto časť sa zaoberá návrhom a tvorbou schémy zapojenia a následným návrhom samotnej dosky plošných spojov, ďalej len ako „DPS“

### 2.1 Altium Designer

Altium Designer je návrhový systém pre elektroniku, ktorý oproti programom ako Eagle alebo KiCad, integruje väčšinu činností návrhu do jedného programu a umožňuje tak lepšiu previazanosť čiastkových úloh pri samotnom návrhu a tým i lepšiu kontrolu nad samotným projektom. Tvorcom programu je austrálska spoločnosť Altium Ltd., a samotný program naväzuje na predchodcu menom Protel. Program je však viazaný pomerne drahou licenciou, avšak pre študentov vysokých škôl je dostupný vo forme ročnej študentskej licencie, ktorá je plnohodnotnou verziou. Altium patrí medzi profesionálne nástroje pre tvorbu nielen jednoduchších DPS ale umožňuje návrh CPU, konfigurácie obvodov FPGA a CPLD, poskytuje taktiež vlastné obvodové simulácie, ktoré nahrádzajú programy ako PSpice a Microcap, zároveň dokáže tvoriť jednoduché GUI pre daný obvod slúžiace pre ovládanie daného obvodu. Obsahuje i vlastný 3D návrhový systém pre lepšiu vizualizáciu výslednej DPS, poprípade i s osadenými súčiastkami.

Pracuje s vlastnými knižnicami „IntLib“, ale vďaka prepojeniu s online databázou je možné priamo v programe vyhľadávať daný komponent od daného výrobcu a zároveň poskytuje odkaz na dostupnosť u predajcov ako sú Farnell, Mouser, TME a pod. Obsahuje taktiež možnosť tvorby vlastných knižníc schematických symbolov súčiastok, tvorbu ich footprintov a pomocou sprievodcu rýchlu tvorbu 3D modelov daných súčiastok.

Systém poskytuje obrovskú nastaviteľnosť návrhových pravidiel a tak udržiavať návrh v poriadku už počas tvorby, oproti iným programom už nepoužíva maticový systém zadávania pravidiel a výnimiek ale používa prehľadné menu tvorby pravidiel, poskytuje taktiež sprievodcu tvorbou. Výhodou ktorú poskytuje je tzv. „online kontrola – online DRC violations“, ktorá okamžite reaguje na porušenie pravidiel zmenou farby komponentu.[26]

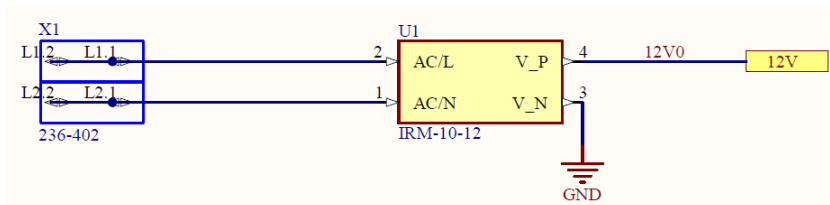
## 2.2 Schéma zapojenia

Časť schéma zapojenia obsahuje postupne všetky dielčie časti zapojenia a ich popis funkcie v obvode.

### 2.2.1 Napájanie

#### Primárny zdroj

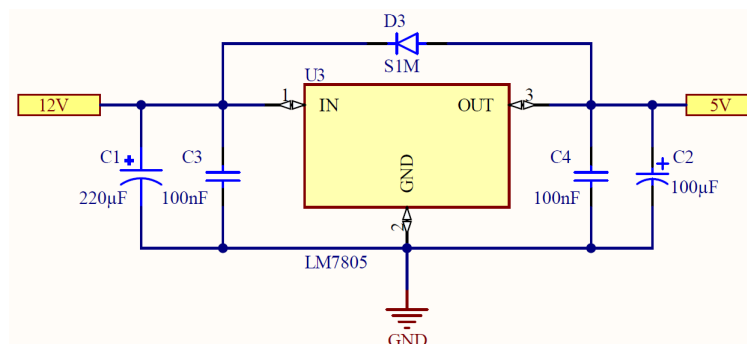
Primárny zdroj elektrickej energie je spínaný sieťový zdroj MeanWell IRM-10-12 , s výstupným napájacím napätím 12 V s maximálnym výstupným prúdom 0,85 A, obsahuje integrované ochrany preťaženia a prepätia. Slúži pre napájanie všetkých komponentov na doske pri normálnom režime napájania a sú z neho odvodzované nižšie úrovne pomocou regulátorov napätia. Zdroj je napájaný z elektrickej siete, ktorá je privedená na dosku pomocou svoriek WAGO 236-402.



Obr. 2.1: Schéma primárneho zdroja

#### Napájacie vetvy

Primárne napájanie je delené ďalej na tri úrovne napájacieho napätia – 12 V; 5 V a 3,3 V. Dvanásť voltová vetva je priama zo zdroja, ostatné vetvy sú vytvárané pomocou stabilizátorov pevného napätia, kde pre päť voltovú vetvu je vyčlenený jeden stabilizátor LM7805-1 A a 3,3 V vetvu sú vyčlenené dva stabilizátory HT7833-500 mA,

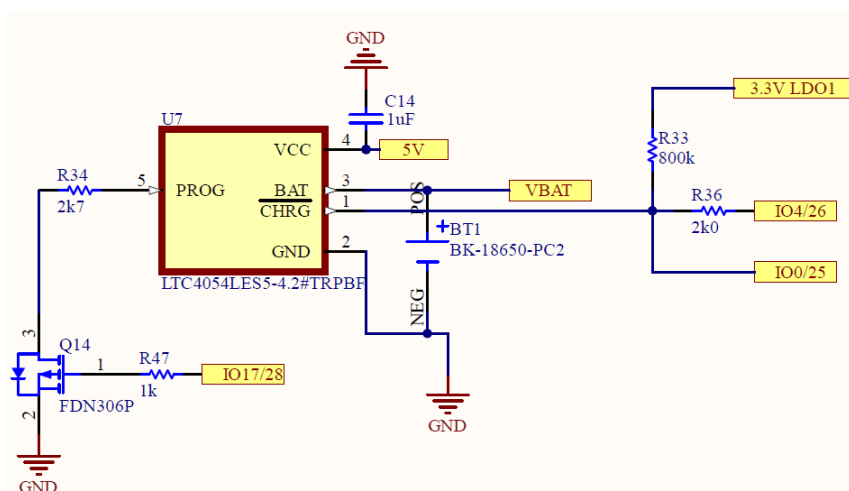


Obr. 2.2: Stabilizátor 5 V

z dôvodu, že prvý je určený pre zálohovanú vetvu cez batériu označený ako *3.3V LDO1* a druhý napája zariadenia, ktoré nevyžadujú zálohu napájania označený ako *3.3V LDO2*. Všetky stabilizátory sú navrhnuté spolu s ich vlastnými filtračnými a blokovacími kondenzátormi, kde medzi vstupnú a výstupnú časť regulátora je priradená dióda *1N4007* v závernom smere tak aby ochránila samotný regulátor pred poškodením, v prípadoch keby sa pri vypnutí na výstupe objavilo väčšie napätie ako na vstupe.

## Sekundárny zdroj

Sekundárny zdroj tvorí, tak ako bolo spomenuté v návrhu, Li-Ion batéria *NCR18650B*, ktorá bude slúžiť pre napájanie *ESP32* spolu s *SD* kartou a obvodom reálneho času pri výpadku primárneho zdroja. Táto batéria pri svojej kapacite dokáže tieto obvody udržať pri „živote“ rádovo desiatky hodín pri prechode *ESP32* do režimu hlbokého spánku, avšak nie je schopná poskytnúť dostatok energie pre chod celého systému. Pre zálohu celého systému by sa musel navrhnuť úplne iný zálohovaný zdroj s omnoho väčšou kapacitou a hlavne s výstupným napätím *12 V*, to však už neprichádza do úvahy pre dané rozmery a účel na danej *DPS*. Celý sekundárny obvod napájania ale nie je len o batérii, ale tvorí komplexnejší obvod, v návrhu sa spomína obvod *LTC4415*, nakoľko je ale tento obvod dlhodobo nedostupný a je potreba prepínať medzi zdrojmi a zároveň musí byť o tom informovaný procesor ako náhrada bol vybraný obvod *LTC4412*[20], ktorý síce neobsahuje všetky funkcie ktoré ponúka *LTC4415* ale je adekvátnou náhradou, ktorá ale vyžaduje prídanie ďalších súčiastok. Súčasťou sekundárneho zdroja je i nabíjací obvod *LTC4054*. Celková schéma sekundárneho obvodu je v prílohe.



Obr. 2.3: Schéma nabíjačky

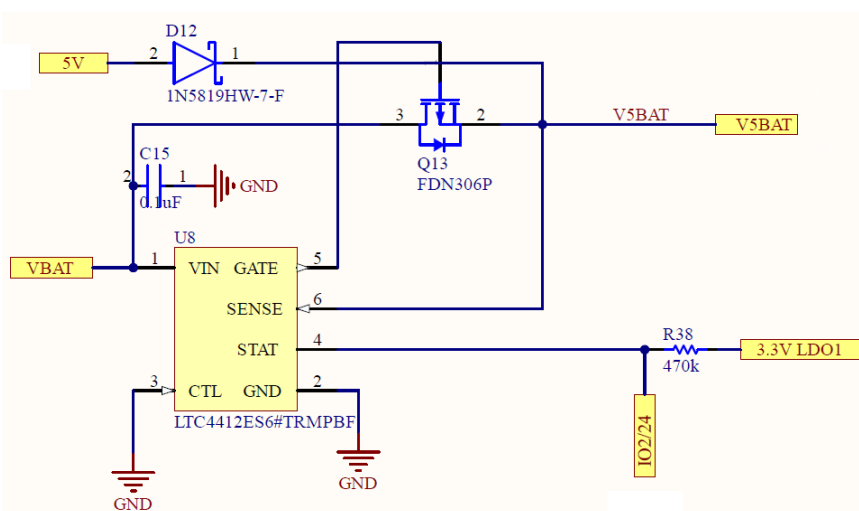
## Popis funkcionality

Päť voltová vetva napája LTC4054, keďže je potrebné vstupné napätie väčšie ako 4,25 V. Nabíjací prúd sa nastavuje pomocou pinu PROG a bol stanovený na 350 mA. Podľa vzorca udávaného výrobcom je výsledná hodnota odporu približne  $2857\ \Omega$ , čo vo výrobnej rade E24 je najbližší možný o hodnote  $2700\ \Omega$ . Čo po prepočte následne vychádza 370 mA. Aby bolo možné riadiť manuálne vypnutie nabíjania, je tento rezistor pripájaný voči zemi pomocou N-MOSFET tranzistoru FDN306P, kedy pri odpojení od zeme klesne nabíjací prúd na  $25\ \mu\text{A}$ . Stav nabíjania je monitorovaný pomocou výstupného pinu CHRG, ktorý sa na mikroprocesor pripojí pomocou dvoch pinov s pull-up rezistormi nastavených jeden ako vstupný a druhý ako výstupný.

Indikujú sa tri stavy:

1. beží nabíjací cyklus, napájacie napätie je v poriadku
2. nabíjanie je ukončené, všetko je v poriadku
3. podpäťové blokovanie, napájacie napätie je nižšie ako 200 mV oproti napätiu na batérii

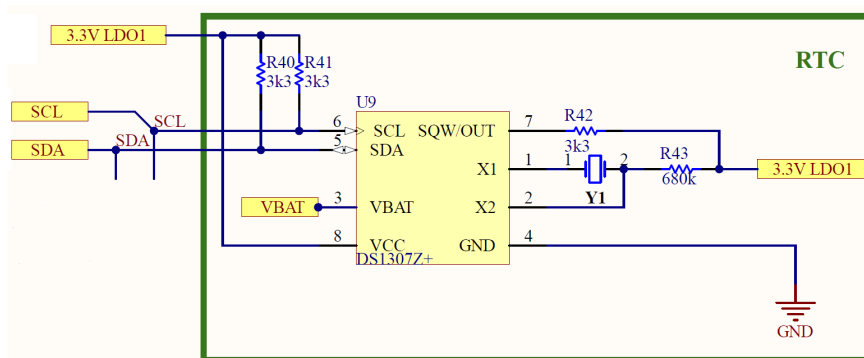
Batéria je ďalej v obvode pripojená na vstupný pin VIN (LTC4412), obvod LTC4412 slúži ako komparátor a prepínač napájania z 5 V vetvy alebo batérie, ale musí byť napájaný z batérie aby sa zabezpečila jeho funkčnosť. Na vstupe SENSE sníma prítomnosť primárneho napätia, rozhodovanie o prepnutí sa ponechá na internom nastavení uzemnením pinu CTL. K prepnutiu dochádza ak primárne napájanie poklesne na úroveň VIN-20 mV, kedy sa pomocou výstupu GATE a tranzistoru FDN306P prepne. Pomocou výstupného pinu STAT je detekované prepnutie na batériu a tak ESP32 môže vykonať potrebné kroky. Obidve napájacie vetvy sú privedené do napäťového regulátora 3.3 LDO1.



Obr. 2.4: Schéma prepínania napájania

### 2.2.2 Obvod reálneho času

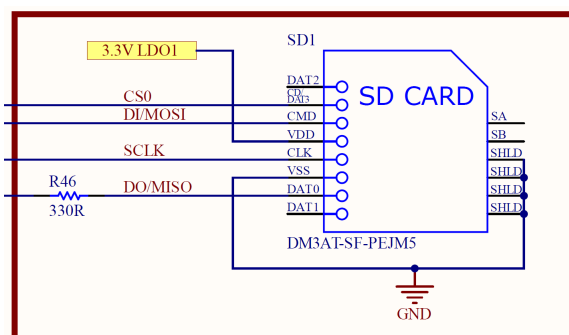
Pre potreby časových značiek je na dosku integrovaný obvod reálneho času DS17307Z. V prvom návrhu sa uvažovalo ako o prídavnom module, ktorý obsahuje ešte malú EEPROM pamäť, avšak nakoniec je priamo integrovaný na doske a EEPROM pamäť supluje SD karta. Taktiež patrí do vetvy zálohovaného napájania, aby neprichádzalo k stratám a de-synchronizácii času a tým pádom k nesprávnym záznamom. Pripojený je cez zbernicu  $I^2C$ .



Obr. 2.5: Schéma RTC

### 2.2.3 SD karta

Na doske sa taktiež nachádza zásuvný modul pre pridanie SD karty, ktorá slúži ako väčšie interné úložisko. Tak ako obvod reálneho času tak i SD karta je napájaná zálohovane, aby na ňu mohli byť ukladané potrebné dáta. Pripojená je priamo cez zbernicu SPI, keďže to ESP32 dovoľuje a nakoľko zálohované napätie je 3,3V nepotrebuje ani SHIFT-LEVEL obvod. Okrem línie DO/MISO, kde pre problémy s interferenciou s RFID modulom musí byť k SD karte pridaný rezistor o hodnote 330  $\Omega$ .

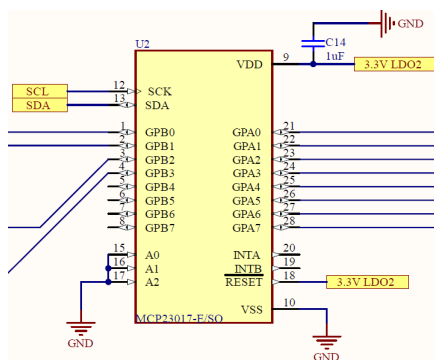


Obr. 2.6: Schéma pripojenia SD karty

## 2.2.4 Ovládanie externých výkonov/ snímanie prítomnosti napätia

### MCP2307

Z dôvodu šetrenia počtu GPIO priamo na ESP32, pri potrebe ponechať si voľné vstupy na ostatné účely, je doska doplnená o nový prvok, a to 16 bitový digitálny obojsmerný expander MCP23017[27]. Poskytuje rozšírenie o ďalších 16 GPIO portov, pracuje na zbernici  $I^2C$  (100, 400 kHz alebo 1,7 MHz) alebo vo verzii S i na SPI(max. 10 MHz), vo verzii  $I^2C$  je možnosť adresovania pomocou troch hardvérových pinov a dva konfigurovateľné piny prerušenia. Pracovné napätie v rozsahu 1,8 – 5,5 V. Vstupno-výstupné piny sú rozdelené na dva nezávislé registre s označením GPA0–GPA7 a GPB0–GPB7 s maximálnym prúdovým zaťažením 25 mA. Pri potrebe snímať 4 digitálne vstupy a ovládať 8 digitálnych výstupov je úspora značná.



Obr. 2.7: Schéma MCP23017

### Snímanie vstupov

Pre detekciu prítomnosti napätí vo výrobnom zariadení je systém vytvorený pomocou opačného zapojenia štyroch relé, v zmysle tak, že typ relé resp. cievky relé je vyberaná podľa snímaného napätia. Výstupy z relé sa pripájajú na dosku pomocou 12-násobnej svorkovnice WAGO 236-412 označenej ako X3. Spoločný kontakt relé(COM/K,common) je privedený cez rezistor na anódu optického oddelovača PC817A[25]. Kontakt, ktorý je v normálnom stave zatvorený(NC/normally close), je privedený na zem a v prípade neprítomnosti daného napätia je vstup optodeľiča pripojený na zem. Na kontakt, ktorý je v normálnom stave otvorený(NO/normally open), je privedené napätie, ak sa objaví na relé cievky napájanie, tak je prepnutím kontaktu aktivovaný optodeľič, ktorého stav prečíta MCP23017. Takto je možné detekovať prítomnosť ľubovoľných štandardizovaných napätí. Vstupy sú pripojené na B register 0–3.

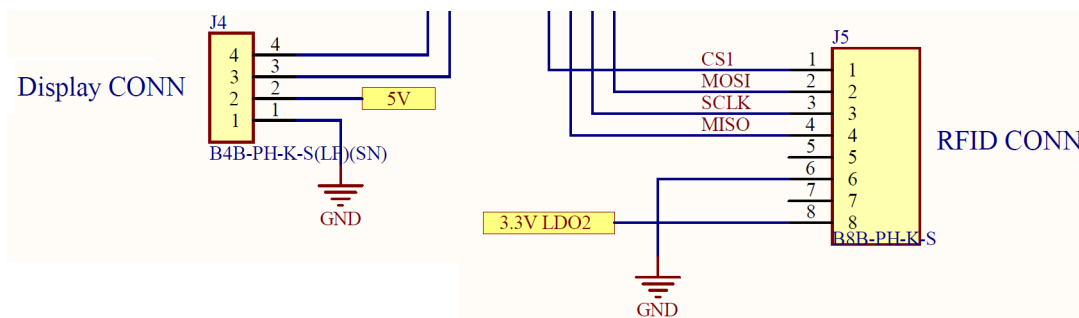


## 2.2.6 Externé periféria

Okrem priamo integrovaných obvodov a súčiastok, je potrebné pripojiť externé periféria, ktoré nie je možné z montážnych dôvodov a osadenia dosky v rozvádzači integrovať na dosku. Do tejto kategórie spadá:

- **Displej** – pripojený pomocou zbernice  $I^2C$ , pripojenie na zbernicu spolu s napájaním je vyvedené na 4 kolíkový JST konektor
- **RFID čítačka** – pripojenie pomocou zbernice SPI, pripojenie na zbernicu s napájaním je vyvedené na 8 kolíkový JST konektor
- **Indukčný snímač** – pripojenie vyvedené na 3 kolíkový JST konektor

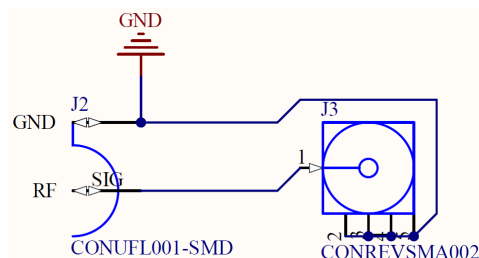
Výber JST konektorov je odôvodnený mechanickým prevedením konektorov, bráni nesprávnemu natočeniu pri pripájaní a zároveň obsahujú mechanický zámok pre zablokovanie náhodného odpojenia konektora.



Obr. 2.9: Schéma pripojenia indukčného RFID a displeju

### Externá anténa

Síce ESP32 disponuje konektorom pre pripojenie externej antény pomocou konektoru U.FL, ale vzhľadom na „jemnosť a miniatúrnu povahu“ konektoru, je pripojenie antény prevedené už na doske z U.FL na SMA konektor, ktorý poskytuje omnoho väčšiu mechanickú odolnosť pri manipulácii.



Obr. 2.10: Schéma pripojenia externej antény



## 2.2.7 Rozloženie na DPS

Plošné spoje sú vytvorené na obojstrannej doske vyrobenej z klasického materiálu FR4 s hrúbkou medi 1 oz, napájacie vetvy sú vytvárané posilnenými spojmi o hrúbke 0,75 a 0,5 mm, signálové cesty sú vytvorené hrúbkou 0,3 mm. Súčiastky budú osadené kombinovanou formou, časť súčiastok – hlavne rezistory, kondenzátory a integrované obvody sú osadené formou SMD. Spínaný zdroj, svorkovnice a konektory (mechanické prvky), opto-oddelovače sú osadené formou THT.

Prívod striedavého napätia 230 V je umiestnený v ľavom hornom rohu dosky s krátkou vzdialenosťou ku vstupným kontaktom zdroja. Prívody sú zároveň po osadení zdroja z veľkej časti kryté púzdom zdroja pre minimalizovanie možného nebezpečného dotyku. Zároveň týmto umiestnením sa maximalizuje vzdialenosť ostatných komponentov od striedavého napätia, hlavne anténnej časti ESP32.

Držiak pre batériu je umiestnený vpravo od zdroja na hornom okraji dosky, pre lepší prístup pri vkladaní alebo vymieňaní batérie, zároveň mechanicky chráni ESP32, ktorá je umiestnená priamo pod púzdom batérie, v približnom strede dosky, tak aby sa napájacie ale i signálové spoje dali viesť čo najjednoduchšie k ESP32. Samotná ESP32 nie je zatiaľ priamo integrovaná na dosku, pre skúšobné účely sa využíva vývojová verzia ES32-WROOM-32U s externým vývodom pre anténu. Tento modul disponuje klasickými vývodmi s roztečom 2,54 mm. Preto sa môže priamo spájať na dosku alebo na dosku osadiť rám s kontaktným poľom pre rýchle nasadenie a prípadnú výmenu. Na pravej strane dosky sa nachádzajú konektory pre vývod externých periférií, i pomocné konektory pre externú anténu a púzdro pre micro SD kartu, tak aby sa dala karta pohodlne vsadiť.

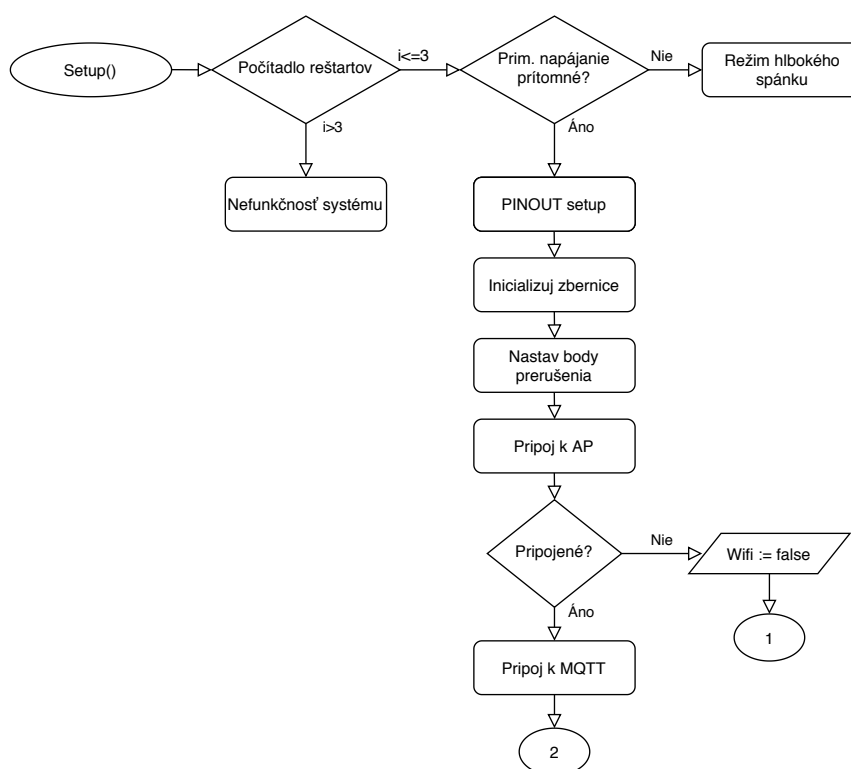
V ľavej časti pod spínaným zdrojom sa nachádza časť s regulátormi napätia, obvod reálneho času a časť pre ovládanie vstupov/výstupov (ULN2803, MCP23017), keďže MCP23017 pracuje na totožnej zbernici s RTC, tak sú umiestnené spoločne pre kratšie prepoje zbernice. Smerom nadol nasleduje časť s opto-oddelovačmi, kde na spodnom okraji dosky sú v jednej línii umiestnené WAGO svorkovnice, a to zľava 12-násobná pre vstupy a následne 16-násobná pre výstupy, ktoré vyžadujú veľký priestor vzhľadom na svoju konštrukciu. Celú schému rozloženia je možné vidieť na obrázkoch v prílohách B.1, B.2 a B.3.

## 2.3 Programové vybavenie

Časť programové vybavenie obsahuje popis funkcií a postupov pre beh celého systému. Prvá časť obsahuje programové vybavenie z pohľadu ESP32 pri dodržiavaní paradigmy Arduino IDE, kde sa využíva zaužívaný systém Setup() a Loop().

### 2.3.1 Setup()

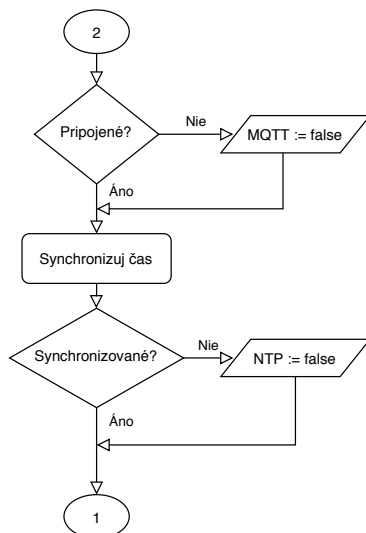
Ako úplne prvý krok pred všetkými nasledujúcimi úkonmi je potrebné zistiť prítomnosť primárneho napájania, pokiaľ je prítomné pokračuje sa ako je to písané v tejto časti ďalej, ak nie je, tak systém prechádza do režimu spánku, ktorý bude opísaný v neskorších častiach.



Obr. 2.11: Vývojový diagram štartu časť 1

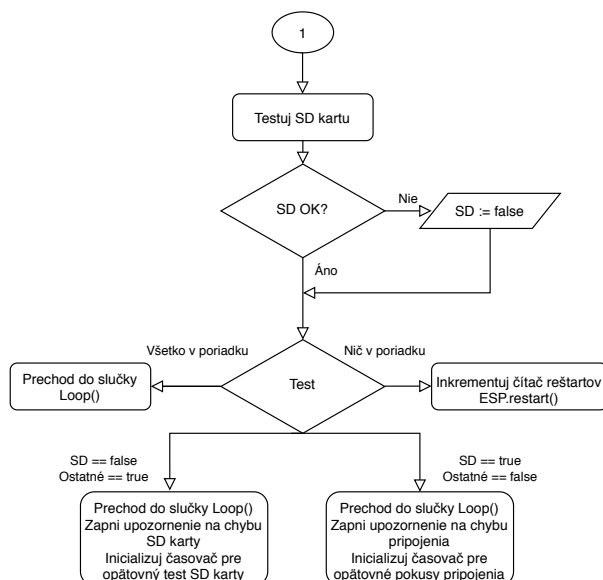
Podstatnou časťou pri nábehu systému je ako prvé potrebné aktivovať zbernice, inicializovať zariadenia (RFID, displej), nastaviť potrebný PINOUT, určiť piny, ktoré budú podliehať systému prerušenia, pripojiť sa na AP, získať potrebné adresy. Následne sa spojiť s MQTT brokerom a overiť funkčnosť komunikácie. Zároveň je potrebné overiť systémový čas a pokiaľ nie je v poriadku, tak ho zosynchronizovať pomocou prijatého času z NTP serveru ako vidno vo vývojovom diagrame na obrázku 2.11.

Následne overiť prítomnosť a funkčnosť SD karty. Samozrejme tieto úkony nesmú podmieňovať ďalší beh programu, preto musia byť vo forme „non-blocking“, tak aby sa pri ich neúspechu pokračovalo ďalej v nábehu systému a o ich vykonanie sa pokúsil systém v neskorších fázach programu.



Obr. 2.12: Vývojový diagram štartu časť 2

Neúspešné pripojenie na AP/do brokeru, nepreverenie času nepatria do kritických segmentov fungovania zariadenia a celý zápis výrobných dát sa presunie na SD kartu do času, kým sa neobnoví komunikácia. U času sa v danom momente musí systém spoliehať, že čas poskytovaný z obvodu RTC je správny.



Obr. 2.13: Vývojový diagram štartu časť 3

Problematické je však zlyhanie SD karty, viď. obrázok 2.13, v danom momente sa systém spolieha len na online prenos dát, identifikácia zamestnanca sa musí vyžiadať z databázy avšak systém môže pokračovať ďalej. Ak príde k zlyhaniu oboch eventualít musí sa vyhlásiť stav nefunkčnosti a systém nesmie pokračovať ďalej a musí prebehnúť celá inicializácia odznova pomocou reštartu, avšak tento reštart je obmedzený pomocou počítadla, ktorý dovolí vykonať reštart len 3 trikrát, ak ani po troch reštartoch nič nepracuje správne systém odchádza do režimu spánku s výpisom chyby na obrazovku. Ak je však všetko v poriadku, poprípade nenabehli nekritické súčasti, prichádza k ukončeniu časti Setup() a prechod do vykonávacej slučky Loop().

### 2.3.2 Body prerušenia - Interrupts

Body prerušenia slúžia na obsluhu resp. reakcie na aktuálne situácie. ESP32 ponúka možnosť vyvolať bod prerušenia pomocou externého vplyvu alebo interného časovaču.

#### Počítadlo pracovných cyklov

Výstup z opto-oddelovača je privedený na vstup ESP32, na ktorý je aplikované priradenie prerušenia s použitím vnorenej lambda funkcie, aby sa nemusela volať obslužná funkcia s príznakom IRAM\_ATTR.

```
pinMode(interruptPin, INPUT);

attachInterrupt(digitalPinToInterrupt(interruptPin), []
{if (ButtonPressed+= (micros() - DebounceTimer) >= (delayTime )) {
    portENTER_CRITICAL_ISR(&mux);
    DebounceTimer = micros();
    portEXIT_CRITICAL_ISR(&mux);
    InterruptCounter++
}}, HIGH);
```

Pre započítanie cyklu je ale vytvorený priamy kód v slučke, ktorý si kontroluje hodnotu ButtonPressed, a to z dôvodu aby kritická časť prerušenia bola čo najkratšia a nezaťažovala sa ďalšími zápismi. Pomocou tejto časti kódu sa do konzoly vypisuje údaj o prerušení pre vlastný debug výpis.

```

if (ButtonPressed > 0 ) {Serial.println("Interrupt") ;
  ButtonPressed = 0; //
  stlac++;
  Serial.println(stlac);}

```

Prerušenie je aktivované v momente, keď sa na vstupe objaví vysoká úroveň(HIGH), uvažovaná bola reakcia na stav RISING – reakcia na nábežnú hranu. Avšak samotný snímač vykazoval pri spínaní veľké zákmity a dlhý neustálený stav. Tieto parazitické vlastnosti sa kopírovali i cez opto-oddelovač. Vzhľadom na rýchlosť procesoru, boli viacnásobne vyhodnocované ako falošné stavy a mylne zarátané. I keď pri reakcii na vysokú úroveň sa tiež objavovali falošné zápisy a tak aby sa nemusela vytvárať dolná prepust' pre vyhladenie zákmitov, aplikoval sa „Debounce“ výpočet, ktorý imunizuje snímanie na neustálené stavy.

## **Zobudenie 0 - dôvod RFID prerušenie**

Pri normálnej činnosti systému nastáva situácia dlhodobej nečinnosti ako ukončenie pracovného dňa, na danom zariadení nik nevyrába. Preto je nepotrebné aby systém fungoval v plne aktívnom režime a tak sa využíva princípů rôznych módov spánku. V tomto prípade je po 2 hodinách od posledného prerušenia od počítačľa cyklov alebo hodinu od posledného odhlásenia zamestnanca (reakcia na posledné prerušenie od snímača cyklov bolo pridané na základe možnosti, že zamestnanec sa zabudne odhlásiť). uvedený systém z dôvodu nečinnosti do módu ľahký spánok. Kedy prichádza k vypnutiu modemu a vypnutiu periférií, okrem toho, ktorý sa namapuje ako bod pre zobudenie zo spánku. V tomto prípade je to prerušenie pomocou prítomnosti RFID tagu, nakoľko MFRC522 disponuje pomocným výstupom IRQ signalizujúcim prítomnosť tagu. Zobudenie zo spánku znamená opätovná prítomnosť pracovníka. V režime ľahkého spánku klesá spotreba procesora na 0,8 mA.

Poznámka: Čip MRFC522 disponuje síce IRQ pinom, ktorý generuje signály pre prerušenie, ale je ich schopný generovať len ak je zbernica SPI aktívna a do registrov prišlo k zápisu hodnôt vymazania predchádzajúceho prerušenia a nového refresh nastavenia. V aktívnom režime prerušenie pracuje spoľahlivo, aj napriek mnohým názorom, že pin IRQ je fiktívny výstup na čipe MRFC522 a zápis do jeho registra je zbytočný.

```

rfid.PCD_Init();
rfid.PCD_WriteRegister(MFRC522::ComIrqReg,0x87);
rfid.PCD_WriteRegister(MFRC522::ComIEnReg,0xA0);
rfid.PCD_WriteRegister(MFRC522::DivIEnReg,0x14);

pinMode(15,INPUT_PULLUP);

attachInterrupt(digitalPinToInterrupt(15),isr,FALLING);
print_wakeup_reason();
rtc_gpio_pullup_en(GPIO_NUM_15)
gpio_wakeup_enable(GPIO_NUM_15, GPIO_INTR_HIGH_LEVEL);
esp_sleep_enable_gpio_wakeup();
esp_light_sleep_start();

```

Pri ľahkom spánku bohužiaľ dochádza k odstaveniu zbernice SPI, jedinou zbernicou možnú mapovať do spánku je UART. Tým dochádza k znefunkčneniu daného prerušovacieho výstupu a dochádza k zamedzeniu jeho použitia ako budiaceho dôvodu (v kóde pre ESP32 označované ako WAKE UP Reason). Preto celý postup ľahkého spánku bude upravený takto: Systém sa v ľahkom spánku považuje za funkčný, a tým pádom si je možné dovoliť v určitých intervaloch zobúdzat procesor. Uvedenie do spánku zostáva nezmenené, takže stále musia byť splnené podmienky posledného odhlásenia a posledného prerušenia od indukčného snímača, avšak toto porovnávanie nastane až v čase po pracovnej dobe. Procesor sa bude zobúdzat v pravidelných hodinových intervaloch, nastavených interným časovačom, takže zdrojom prebudenia sa stane interný časovač. Po zobudení sa vždy skontroluje čas poskytnutý od RTC. A pokiaľ bude čas v časti hodín rovný  $T_{\text{začiatokpracovnejdoby}} - 1$ , systém zostáva zobudený, ak nie tak systém sa znova vráti do režimu ľahkého spánku.

Tým sa zabezpečí nábeh systému minimálne hodinu pred začiatkom pracovnej doby. V pracovnej dobe bude tiež možné vojsť do režimu ľahkého spánku ak by náhodou v daný deň nikto na zariadení nevyrábal, ale v čase pracovnej doby síce bude porovnávaná podmienka nečinnosti, pre uvedenie do spánku ale nebude vykonávané pravidelné zobudenie a bude sa čakať na stlačenie tlačidla na rotačnom enkodére ako prezentovanie prítomnosti pracovníka.

## Zobudenie 1 - obnova primárneho napájania

Pri výpadku primárneho napájacieho napätia, je o tom procesor informovaný z výstupu LTC4412 na pine STAT. Pri zmene na danom vstupe sa preruší vysielanie modemu, posledné údaje sa uložia na SD kartu. Pri výpadku teda systém prechádza do režimu hlbokého spánku, z ktorého ho môže prebudiť jedine návrat primárneho napájania. V móde hlbokého spánku je v nečinnosti modem, všetky periférie (okrem mapovaného), vyradené je i výpočetné jadro ESP32. Zostáva pracovať jedine ULP co-procesor (ULP – Ultra Low Power). ULP má na starosti v pravidelných intervaloch preverovať stav na namapovanom vstupe a buď pokračuje v hlbokom spánku alebo pri zmene stavu zobudí systém do plnej prevádzky. Treba si však dať pozor, nakoľko dochádza k odpojeniu i hlavnej pamäti a zostáva v prevádzke len malá vyhradená časť do ktorej sa ukladá s príznakom RTC. Takže pred odchodom do spánku je lepšie všetky vlastné premenné preuložiť na SD kartu, nakoľko zobudenie zo spánku začína resetom. Prúdový odber v tomto stave sa pohybuje na úrovni 10 – 20  $\mu$ A.

Pri prítomnosti primárneho napájania je na výstupe pinu STAT vysoká úroveň(HIGH), zmenu kontroluje funkcia prerušenia a pri zmene na nízku úroveň, čo značí stratu primárneho napájania, sa aktivuje funkcia prerušenia, ktorá korektne odpojí Wifi, pre-uloží priebežné dáta na SD kartu a uvedie systém do režimu hlbokého spánku. V prípade tejto funkcie sa nebude volať s príznakom IRAM\_ATTR, nakoľko tento krok nie je časovo kritický a zároveň sa tým zbytočne nenavýši spotreba. Následne po prechode do režimu hlbokého spánku je ten istý vstup mapovaný ako bod pre obnovu do aktívneho režimu, kedy sa sleduje opätovná prítomnosť vysokej úrovne. Informovanie MQTT brokeru neprebehne, nakoľko ak sa do 30 sekúnd stanica neprihlási pomocou PING príznaku, alebo neodošle dáta, tak príde k vyhláseniu straty stanice automaticky.

```
pinMode(2, INPUT);
attachInterrupt(digitalPinToInterrupt(2), vypadok, LOW);

void vypadok(){
    WiFi.disconnect();
    WiFi.mode(WIFI_OFF);
    saveSDdata();

    esp_sleep_enable_ext0_wakeup(GPIO_NUM_2, 1);
    esp_deep_sleep_start();
}
```

### 2.3.3 Pripojenie k AP a MQTT brokeru

Vzhľadom na to, že zariadenie bude umiestnené stabilne na určenom mieste a pripájané do podnikovej siete s vlastnou podsietou, nie je potreba získavať IP adresy pomocou protokolu DHCP, zadávať ani adresy DNS. Zo strany ESP32 bude pohľad do siete výrazne obmedzený na komunikáciu len s vlastným serverom a NTP serverom. Sprostredkovanie komunikácie s užívateľom a užívateľa s ESP32 smerom z internetu bude obsluhovať broker. Ďalším faktom sú problémy s DHCP na strane ESP32, ktorá je programovaná cez Arduino IDE, chyba sa prejavuje v neustálej a opakovanej strate prijatých adries a tým si ESP neustále vyžiadáva adresy opakovane i keď nevypršal čas platnosti pridelenia. Pravdepodobne ide o chybu knižnice, ale na stránkach vývojárov sa píše, že ide o dlhodobý problém v prých verziách ESP32 a užívatelia sa s ňou stretávajú i keď programujú cez iné prostredia než Arduino IDE. Čo skôr teda značí chybu priamo pri implementácii IP stacku. Pre jednoznačnú identifikáciu i v sieti je lepšie v tomto prípade priradiť adresu statickú. Zároveň statickú adresu dostal i broker Mosquitto, preto je jeho adresu možné zadať priamo v kóde. Pre využitie funkcionalít Wifi jadra sa volá knižnica „Wifi.h“, pre nasadenie a využívanie protokolu MQTT sa používa knižnica „PubSubClient.h“. V kóde sa zatiaľ nachádza interná domáca sieť, nakoľko testy a server je nasadený lokálne na vlastnom počítači.

```
const char* ssid      = "borsuk_satro";
const char* password = "*****";
const char* mqtt_server = "192.168.16.101";
```

```
WiFiClient espClient;
PubSubClient client(espClient);
```

Inštancie sú vytvorené v náväznosti, tak aby jeden klient prevzal kontrolu nad riadením Wifi a zároveň pod neho spadala komunikácia na protokole MQTT. V Setup() časti sú nasledne volané funkcie pre pripojenie k Wifi AP, priradenie adresy serveru a portu. Kde funkcia callback overuje naviazanie komunikácie s brokerom, resp. overenie správy.

```
wificonnect();
client.setServer(mqtt_server, 1883);
client.setCallback(callback);
```



### 2.3.4 Identifikácia zamestnanca

Zamestnanec sa pred začatím práce prezentuje priložením RFID tagu k čítačke. Načítaná adresa tagu sa porovná zo záznamom na karte a priradí sa k nemu meno, zároveň sa vytvorí záznam o prihlásení zamestnanca s časovou značkou. Opätovným priložením tagu sa zamestnanec odhlási a znova sa vytvorí záznam o odhlásení. Ak sa však zamestnanec zabudne odhlásiť, tak po nečinnosti systému sa zamestnanec odhlási automaticky. Popri tom sa odošle informácia o prihlásení daného zamestnanca do brokeru. Pokiaľ sa na karte nenájde záznam, porovnanie s menom sa vyžiada z databázy.

### 2.3.5 Záznamy na SD karte

Na SD karte sú udržiavané rôzne súbory záznamov, ktoré popri odosielaní na server pomáhajú vytvárať zálohu dát a udržiavať prehľad o funkčnosti a živote zariadenia:

- **employers** – súbor udržiava zoznam RFID identifikátorov priradených k menám zamestnancov, prehliada sa ako prvý a tým odľahčuje požiadavky do siete
- **activity** – záznamy s časovou značkou o prihláseniach a odhláseniach
- **cyday** – záznamy o počte denných cyklov s priradenými menami a dobou trvania
- **cytotal** – záznamy o stave všetkých doteraz vykonaných cykloch s kompletným prevádzkovým časom
- **servis** – záznamy o servisoch s časovou značkou, časom trvania a menom servisného zamestnanca
- **technical** – kódové záznamy o udalostiach, ktoré sa vyskytli počas behu s časovou značkou
- **temporary** – pri prechode do režimu spánku sa do súboru preuložia všetky potrebné premenné pre opätovný nábeh systému, tak aby pokračoval tam, kde predtým musel skončiť

Pre rýchly a jednoduchší systém zápisu udalostí bolo vytvorené kódové označenie udalostí a ukladajú sa len časové značky s kódom udalosti:

- |  |                                |
|--|--------------------------------|
| • 00x – Napájanie  | • 01x – Wifi                   |
| 000 – All OK   | 010 – Wifi OK                  |
| 001 – výpadok napájania                                  | 011 – výpadok                  |
| 002 – obnova napájania                                   | 012 – obnova                   |
| 003 – výpadok napájania, sekundár príliš nízko, nenabíja | 013 – signál OK, AP neodpovedá |
| • 02x – MQTT   | • 03x – Snímané napätia        |
| 020 – OK   | 030 – All OK                   |
| 021 – Wifi OK, MQTT Err                                  | 031 – Err 1                    |
| 022 – obnova   | 032 – Err 2                    |
|  | 033 – Err 3                    |
|  | 034 – Err 4                    |
| • 04x – Batéria, nabíjanie                               | 043 – BATT Critical            |
| 040 – BATT OK/discharging                                | 044 – BATT Dead                |
| 041 – BATT OK/charging                                   | 045 – UVLO                     |
| 042 – BATT LOW/discharging                               | 046 – BATT Charging End        |

Nie je vylúčené, že typy a množstvo záznamov sa bude meniť, jedná sa o prvý komplexnejší popis udalostí, ktoré je vhodné zaznamenávať a časom určite pribudnú udalosti, ktoré sa ukážu ako potrebné zaznamenávať.

## Práca s SD kartou

Pre prácu so súbormi je potrebné pridať knižnicu pre prácu s SD kartou, so súbormi a súborovými systémami a prácou s textom (FS.h, SD.h, string.h a stdio.h). Pre samotnú prácu je potrebné vytvoriť inštanciu „File“ pre priamu prácu so súbormi. V časti Setup() je potrebné na zbernici SPI inicializovať linku pre SD kartu s výberovým pinom (Chip Select – CS), pre SD kartu je priradený GPIO 5, označený ako CS0, ide o priamy hardverový výstup zbernice SPI, ostatné sú softvérovo emulované. Chybou ESP32 musí byť vždy SD karta riadená hardvérovou verziou. Po inicializácii zbernice je potrebné overiť prítomnosť karty. Následne, ak je prítomná, je vhodné identifikovať typ(MMC, SD, SDHC), následne overiť formátovanie karty(doporučené - FAT32 – 32 kB). Karta musí byť už naformátovaná externe, ESP32 nie je schopné riadiť formátovanie. Týmito úkonmi je overené, že karta je správne identifikovaná, v poriadku a pripravená na prácu. Následne sa ešte overí veľkosť karty a množstvo zapísaného a voľného priestoru.

```
if(!SD.begin(5)){
Serial.println("Chyba zbernice");
SDCARD=false; return;}
uint8_t cardType = SD.cardType();
    if(cardType == CARD_NONE){
        Serial.println("Karta neprítomná");
        return;}
Serial.print("Typ: ");
    if(cardType == CARD_MMC){
        Serial.println("MMC");
    } else if(cardType == CARD_SD){
        Serial.println("SDSC");
    } else if(cardType == CARD_SDHC){
        Serial.println("SDHC");
    } else {
        Serial.println("UNKNOWN");
    }

uint64_t cardSize=SD.cardSize()/(1024*1024);
Serial.printf("Veľkosť:%lluMB\n", cardSize);
Serial.printf("Maximálna dostupná:%lluMB\n",SD.totalBytes()/(1024*1024));
Serial.printf("Využitie miesta:%lluMB\n",SD.usedBytes()/(1024*1024));
```

## Súbory

Pre prácu so súbormi sú vytvorené funkcie:

- tvorbu adresárov
- prehliadanie adresárov
- tvorba súborov
- vymazanie súboru
- vymazanie v súbore
- zápis do súboru
- pridávanie do súboru
- výpis súboru
- hľadanie v súbore
- premenovanie súboru

### 2.3.6 Napájanie, nabíjanie

#### Stav primárneho napájania

Stav primárneho napájania je možné, vďaka zapojeniu LTC4054 a LTC4412, sledovať nielen na prítomnosť ale aj na doplnkový stav. Doplnkovým stavom sa myslí, možnosť detekcie poklesu primárneho napájania pod úroveň napätia na batérii. Ako bolo spomenuté LTC4412 poskytuje výstup STAT. Ale obvod LTC4054 disponuje signalizáciou UVLO, pri ktorej blokuje nabíjanie. Tým pádom je možné získať informáciu o nevyhovujúcom stave primárneho napájania, čo môže značiť chyby zdroja, chyby stabilizátora, poprípade nekorektný prúdový odber spôsobujúci pokles napätia. Ako bolo opísané v sekcii prerušenie, výpadok napájania sa detekuje zmenou hodnoty a reakciou bodom prerušenia. UVLO je detekované len klasickým pravidelným zisťovaním stavu a je opísané v časti Kontrola nabíjania.

#### Stav batérie

Stav napätia na batérii je zisťovaný klasickým AD prevodom. Vzhľadom na to, že na Li-Ion sa objavuje napätie do 4,2 V, a ESP32 dokáže na AD prevodníku čítať maximálne do 3,3 V. Disponuje síce v nastaveniach možnosť nastaviť útlm, a tak čítať i vyššie hodnoty, ale je zbytočné takto zatažovať vstup, keď je možné tento problém vyriešiť klasickým deličom napätia. Delič tvoria dva rezistory s hodnotami 27 a 100 kΩ. To znamená, že pri takto nastavenom deliacom pomere sa pri hodnote 4,2 V objaví na výstupe napätie 3,3 V. Následne je vytvorený prepočet vychádzajúci z 12-bitového rozlíšenia prevodníka:

$$\text{Rozlíšenie} = \frac{3,3 \text{ V}}{4095}, \quad (2.1)$$

$$\text{Rozlíšenie} = 0,85 \text{ mV}, \quad (2.2)$$

kde pri maximálnej úrovni napätia 12-bitový SAR prevodník poskytuje 4095 úrovní.

Následne kód pre prepočet započítava pomer medzi reálnym a meraným napätím:

$$\text{Pomer} = \frac{4,2\text{ V}}{3,3\text{ V}}, \quad (2.3)$$

$$\text{Pomer} = 1,2727, \quad (2.4)$$

čo znamená, že reálne rozlíšenie je tiež násobené týmto pomerom,

$$\text{Rozlíšenie}_{real} = 0,85\text{ mV} \times 1,2727 \quad (2.5)$$

$$\text{Rozlíšenie}_{real} = 1,05\text{ mV}. \quad (2.6)$$

Takže i merania zatažené váhou prepočtu sú naďalej dostatočne presné, a rozlíšenie v ráde jedného milivolta je vysoko dostačujúce.

### Manuálne riadenie nabíjania

V programe je možné ručne zablokovat nabíjanie pomocou odpojenia rezistoru pripojeného na riadiaci pin PROG(LTC4054), ktorý je voči zemi spínaný MOSFET tranzistorom. To dáva možnosť mať pod kontrolou prúdový odber resp. rýchle zníženie prúdového odberu.

### Zisťovanie stavu nabíjačky

LTC4054 na výstupe CHRG pomocou interného MOSFET-u dáva možnosť sledovať tri stavy nabíjačky zmenou interného odporu. Väčšina zapojení je však riešená len smerom k indikácii nabíjania, resp. pripojená na tento výstup LED dióda. Táto metóda poskytuje len svetelnú indikáciu – nabíja/nenabíja a neinformovala by procesor o stavoch. Avšak tieto stavy musia byť zisťované manuálne pomocou dvoch GPIO. Prvý je nastavený ako vstupný so spustením čítania napätia pomocou AD prevodu. Druhý ako výstupný a ovláda pridaný MOSFET tranzistor pre odpájanie testovacieho rezistoru, viď. schéma v prílohe. Výsledné stavy sú monitorované takto:

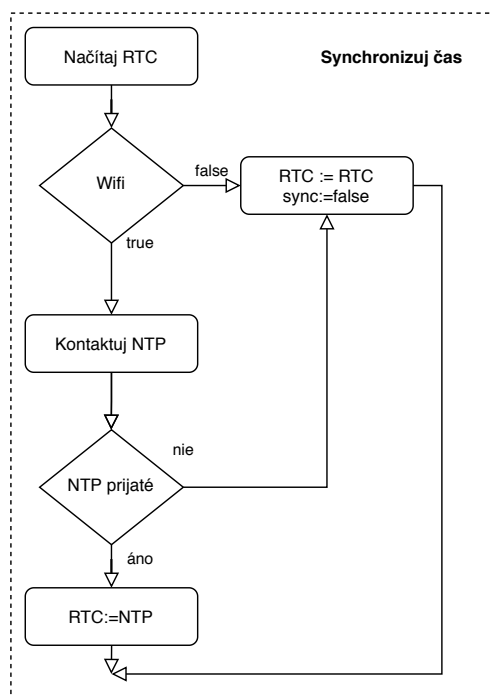
- **Nabíjanie/napájanie OK** – na vstupe(ADC) je merané napätie 1,5 – 1,73 V, externý MOSFET je zopnutý voči napätiu 3,3 V

Pri zmene stavu(koniec nabíjania alebo UVLO) je táto zmena detekovaná zmenou napätia na vstupe na hodnotu 3,3 V. Pre testovanie, ktorá z dvoch situácií nastala je potrebné odpojiť MOSFET. Následne je vyhodnotená situácia:

- **UVLO** – na vstupe(ADC) je merané napätie 1,71 – 2 V
- **Koniec nabíjania/napájanie OK** – na vstupe(ADC) je merané napätie 0 V

## Čas

Čas pre časové značky v klasickom formáte(deň, mesiac, rok, hodina, minúta,sekunda) je poskytovaný primárne z obvodu reálneho času DS1307 pripájaného pomocou zbernice  $I^2C$ . Vzhľadom nato, že tento obvod zdieľa napájanie s ESP32, tak v prípade úplnej straty napájania i z batérie, prestane obvod generovať čas. Na jeho spätné nastavenie je preto vyžiadaný čas z NTP POOL serveru na obnovu času a zároveň slúži na zarovnanie nepresnosti času, nakoľko DS1307 sa vie pomýliť aj o niekoľko minút do mesiaca. Avšak zase spoliehať sa čisto na NTP servery nie je vhodné, v prípade výpadku komunikácie nebude možné získať čas zo serveru a tak sa jediným zdrojom času stane obvod reálneho času.



Obr. 2.14: Vývojový diagram synchronizácie času

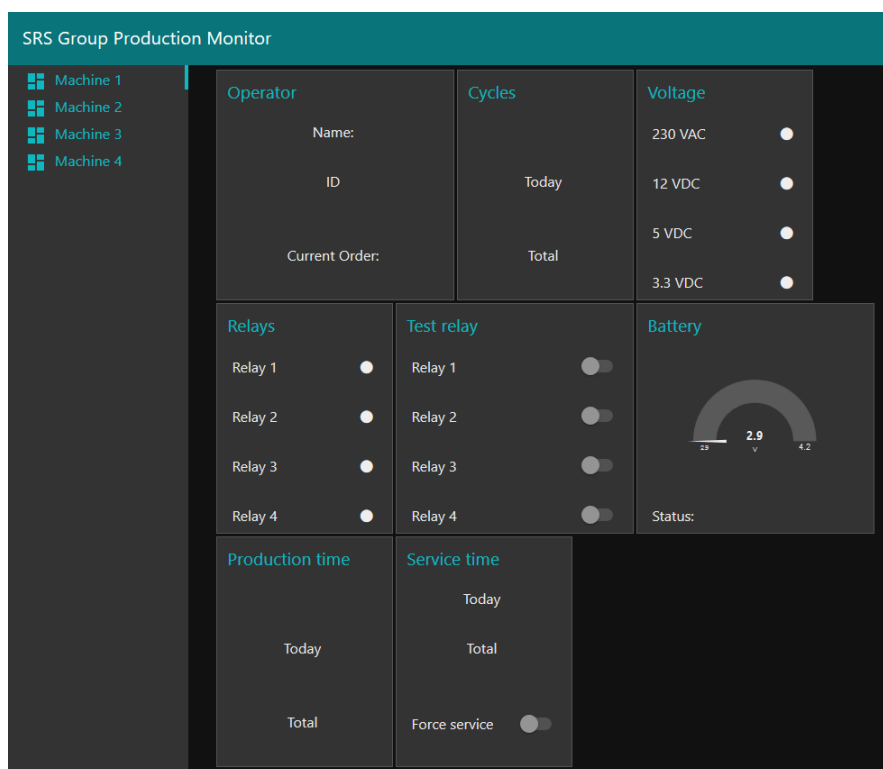
## MCP23017 vstupy a výstupy

Tak ako je ukázané v schéme, nakoniec prišlo ku pridaniu pomocného periféria MCP23017, ktoré je taktiež riadené pomocou zbernice  $I^2C$ . Na túto časť nie sú nasadené žiadne body prerušenia. Prítomnosti napätí sa kontrolujú v pravidelných 10 sekundových intervaloch, čo je dostatočný čas pre zisťovanie zmien. Výstupy sú riadené podľa toho ako sú priradené k udalostiam.

## 2.4 Webové rozhranie

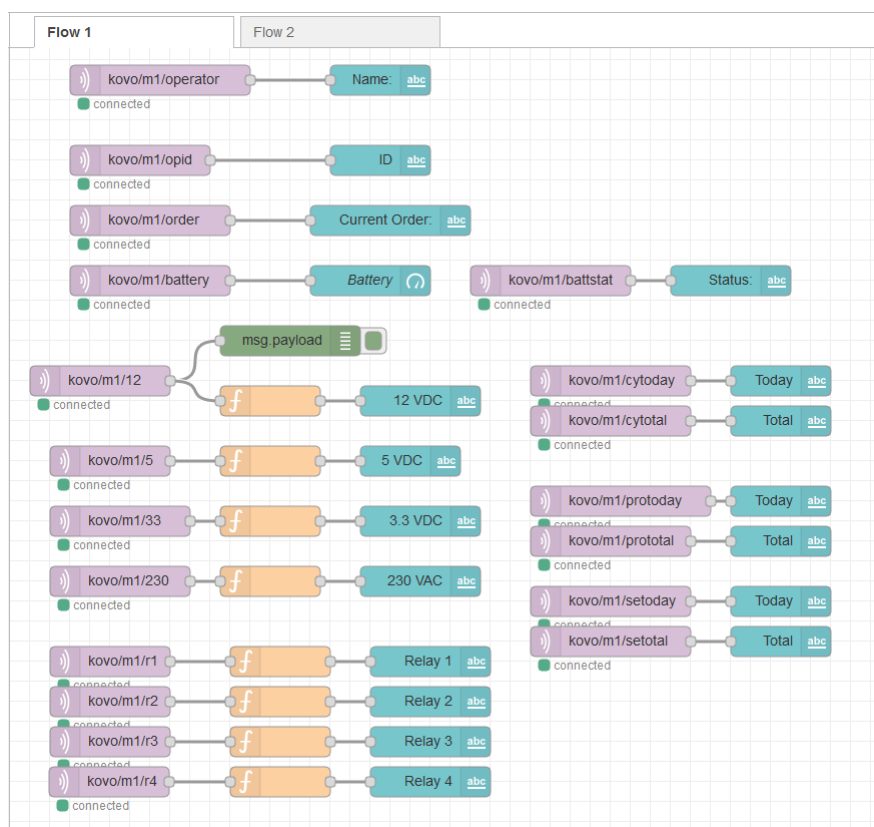
Grafické webové rozhranie je nasadené priamo v NodeRed, nakoľko poskytuje možnosť priamo v ňom vytvoriť webové rozhranie. Poskytuje rýchly prehľad o aktuálnom dianí všetkých zariadení vo výrobe. Obráz je rozdelený klasicky na stĺpcový dvojité layout s menu vľavo. Vpravo sa nachádzajú všetky aktuálne údaje o prevádzke.

V časti „Operator“ je zobrazené meno prihláseného zamestnanca s jeho priradeným firemným ID a priradenú aktuálne rozpracovanú zákazku. V časti „Cycles“ sú informácie o počtoch vykonaných cykloch na zariadení za deň a kompletne cykly, ktoré zariadenie vykonalo. Časť „Voltage“ v novšej verzii označené už ako „Inputs“ poskytuje stav vstupov. Na obrázku 2.15 a v prílohe B.5 (ktorá poskytuje náhľad na „naplnenú“verziu) možno v tejto časti vidieť rôzne napätia, ale to sa môže meniť podľa daného zariadenia, tieto názvy boli vybrané len pre testovacie účely. To isté platí o časti „Relays“ alebo „Outputs“, kde počet použitých výstupov sa bude od zariadenia k zariadeniu líšiť i s názvami. Nasledne je časť, kde dočasne je možné vzdialene spínať výstupy. Nasleduje časť venovaná batérii, kde sa zobrazuje napätie baterky a stav nabíjačky. V časovej časti sa zobrazujú časy dĺžky práce vykonanej za deň a celkovo. To isté platí i o servisnom čase, kde je možné vynútiť servis zariadenia.



Obr. 2.15: Webové/grafické rozhranie

Prostredie je tvorené ako bolo už spomínané v NodeRed, kde na obrázku 2.16 možno vidieť náhľad toho ako vyzerá tvorba vzťahov a ich grafických prvkov s následným vykreslím vid. príloha B.5. Prostredie poskytuje intuitívne prostredie pre rýchlu tvorbu hlavne vďaka pred-pripraveným prvkom, ako sú napríklad SUBSCRIBE položky pre priame získavanie dát z brokeru, PUBLISH položky pre odosielanie na broker. Taktiež možná tvorba funkčných tlačidiel, grafické zobrazovanie hodnôt, tvorba prehľadného menu. Podstatné sú položky pre písanie vlastných funkcií, kde vďaka tomu môžeme časť pracovnej záťaže preniesť z ESP32 na server. I zjednodušiť určité zápisy voči SQL databáze. Dalo by sa nazvať že NodeRed ponúka určitú formu post a pre procesingu.



Obr. 2.16: Tvorba vzťahových funkcií

V ďalšej tvorbe prostredia bude vhodné okrem denného a celo výrobného zobrazenia časov a cyklov doplniť položku, poprípade ďalšiu kartu menu zameranú buď len priamo na zákazku alebo zamestnanca. Nakoľko je s veľkou určitosťou možné povedať, že počas dňa sa vymenia i zákazky i zamestnanec a to i niekoľko krát.



## Záver

Pri tvorbe samotného návrhu sa ukázalo, že treba dôkladne dbať na samotné celkové zapojenie periférií k ESP32, vysoká miera integrácie viacerých funkcií do jedného pinu prináša síce v jednom smere výhody menšieho puzdra samotného čipu ESP32, ale i nevýhody v tom zmysle, že sa musí dbať nato kam budú jednotlivé periférie pripájané, tak aby sa neprišlo o ďalšie funkcie, ktoré môže potrebovať iné rozhranie alebo interná funkcia. Ako napríklad pri 4-bitovom zápise na pamäťovú kartu sa obsadzuje i pin GPIO 2, ktorý ale patrí medzi strapovacie piny a podľa svojho štartovacieho stavu, zodpovedá do akého módu sa ESP32 dostane, či už reštart, nahrávanie programu alebo vykonávanie kódu. Preto je síce obmedzujúce, ale výhodnejšie použiť 1-bitový zápis, aj keď táto forma zápisu je pomalšia. Ukázalo sa, že v tomto prípade je lepšie sa úplne vyhnúť pred-pripravenému SD rozhraniu a SD kartu pripájať klasicky pomocou zbernice SPI.

Ďalej sa ukázalo, ako je ESP32 citlivá na napájanie, pri dlhodobom spánku síce odoberá niekoľko jednotiek mikroampérov, no pri vysielaní TX paketu sa spotreba zvýši až na úroveň 650 mA, čo sú veľké dynamické rozdiely a tomu je potreba prispôbiť i celý obvod napájania. Ďalším aspektom samotného napájania je citlivosť ESP32 na poklesy napätia, kedy pri poklese pod 3,1 V sa chybou hardverovej implementácie okamžite spúšťa watchdog reset, následne sa zdroj zotaví, ESP32 začne znova pracovať a vysieľať, znova klesne napätie a priebeh sa opakuje. Preto aby nedochádzalo k týmto cyklom, je treba zabezpečiť dostatočne stabilný a silný zdroj, čo taktiež platí pri voľbe záložného zdroja a obvodu pre prepínanie zdrojov, tak aby nedochádzalo k zbytočným zákrmitom, čo opäť vedie k nechcenému reštartu. Preto je možné, že príde ešte k zmene celého napájacieho obvodu.

Čo sa týka konečného riešenia napájacieho a nabíjacieho obvodu, LTC4054 bolo ponechané ale prišlo k kámene LTC4415 za LTC4412. Obvod LTC4415 je posledný polrok absolútne nedostupný a nakoniec sa podarilo zohnať jeho dostupnejšiu náhradu LTC4412. Nevýhodou ale je že neobsahuje také funkcie ako novšie LTC4415 a tak prišlo k celkovej úprave obvodu pre prepínanie napájania. Vďaka kaskádovému zapojeniu LTC4054 a LTC4412 vzniklo zapojenie, ktoré vďaka tomu ako majú nastavené komparátory s hystereziou (u LTC4054 je to 100 mV, u LTC4412 200 mV) už pri zablokovaní nabíjania vzniká 100 mV rezerva pre reakciu, že primárne napájanie nie je v poriadku a v blízkej dobe príde k prepnutiu na batériu.

(Ďalej je potrebné systém dovybaviť obvodom automatického resetu a nahrávania, tak aby pri nahrávaní kódu nemuseli byť permanentné stláčané tlačidlá EN a RST.) Nakoniec nebolo potrebné obvod dovybaviť automatickým resetom, nakoľko v aktualizáciách Arduino IDE prišlo k úprave a celé nahrávanie si Arduino IDE riadi samé, takže pokiaľ dĺžka USB kábla nepresiahne 10 metrov, nie je potrebné pridávať 10  $\mu$ F kondenzátor medzi pin EN a 3V3. V konečnom zistení, pri využívaní vývojových modulov ho samotný vývojári začali v novších verziách implementovať.

Z dôvodu nedostatku adekvátnych pinov bol systém oproti prvotnému návrhu doplnený o expander MCP23017, ktorý poskytol ďalších 16 GPIO. Aj keď sa v návrhu rátalo s tým, že ESP32 má dostatok GPIO, ich rozmiestnenie a viacnásobná funkcionálna znižuje použiteľnosť. Na MCP23017 bolo presunuté ovládanie výstupov a snímanie prítomnosti napätí.

Napätia prítomné vo výrobných zariadeniach bude lepšie len detekovať pomocou obvodu s relé s obráteným zmyslom zapojenia, ktoré sa nebudú priamo vyskytovať na DPS ale budú umiestnené na DIN lište v rozvážači. Takto sa vytvorí univerzálny systém pre detekciu, a bude možné ľubovoľne zamieňať relé s danou cievkou. Či už obvody detekcie alebo ovládania zafarčí je vhodné ochrániť pomocou optoizolácie pri poruchách ESP32, pre možnú zmenu napätia cievky bol doplnený obvod ULN2803A, Nakoľko sa predpokladá, že samotné vyrobené moduly budú umiestnené v oceľových rozvážačoch, ktoré by filtrovali bezdrôtový prenos, je vybraná verzia s výstupom na externú anténu, pre ktorú bol na dosku doplnený vlastný SMA konektor.

Konečný PINOUT sa pri vyskladávaní systému nakoniec niekoľkokrát menil a to hlavne z dôvodu využitia nových prvkov. RFID modul sa správal korektne, oproti SD karte, ktorá sa pokiaľ nenachádzala na tej istej napájacej vetve vykazovala chybu pripojenia. Vývojári uvádzajú že ESP32 zvláda i 16 GB karty, ale už pri 8 GB má občas problém s pripojením a identifikáciou.

Ukázalo sa, že pre základnú prácu a základné programovanie je Arduino IDE dostačujúce, avšak implementácia bodov prerušenia ukázala nedostatky v podobe nesprávnej konfigurácie registrov a tým destabilizuje reakciu na prerušenia. Preto je vhodné pri pokročilejších programoch začať využívať programovanie vo vlastnom prostredí ESP-IDF. Nakoniec toto prostredie už ponúka možnosť napísať kód priamo pre ULP co-processor a tak omnoho lepšie riadiť režimy spánku. Za ďalšie by sa tým pravdepodobne podarilo vyriešiť nechcené odpojenie zbernice SPI počas režimu hlbokého spánku.

Výsledný návrh dosky plošných spojov má rozmery 152 x 154 mm, čo sa ešte časom vývoja bude meniť, hlavne bude potrebné zmenšiť jednu zo strán a vytvoriť viac obĺžnikový tvar dosky.

# Literatúra

- [1] PBWCZ. *IoT*. [online]. [cit. 2019-10-14]. Dostupné z URL: <<https://www.pbwcz.cz/elearning/iot.htm>>
- [2] HRUBOŠ, Marián, Peter NAGY a Aleš JANOTA. *Automatizácia – súčasť nášho života*. Veda na dosah: Elektrotechnika, automatizácia a riadiace systémy [online]. Žilina: Katedra riadiacich a informačných systémov; Elektrotechnická fakulta; Žilinská univerzita, 2015 [cit. 2019-10-14]. Dostupné z URL: <<https://vedanadosah.cvtisr.sk/automatizacia-sucast-nasho-zivota>>
- [3] Espressif Systems IOT Team *ESP8266EX Datasheet*. [online]. 2015. [cit. 2019-10-17]. Dostupné z URL: <[https://cdn-shop.adafruit.com/product-files/2471/0A-ESP8266\\_\\_Datasheet\\_\\_EN\\_v4.3.pdf](https://cdn-shop.adafruit.com/product-files/2471/0A-ESP8266__Datasheet__EN_v4.3.pdf)>.
- [4] Espressif Systems IOT Team *ESP32 Series Datasheet*. [online]. 2019. [cit. 2019-10-17]. Dostupné z URL: <[https://www.espressif.com/sites/default/files/documentation/esp32\\_datasheet\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf)>.
- [5] Random Nerd Tutorials *ESP32 Pinout Reference*. [online]. 2019. [cit. 2019-10-18]. Dostupné z URL: <<https://randomnerdtutorials.com/esp32-pinout-reference-gpios/>>.
- [6] BORSUK, Adam. *Automatický hydroponický systém*. Brno, 2018, 88s . [cit. 2019-10-19]. Bakalárska práca. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačných technológií, Ústav telekomunikací. Vedúci práce: doc. Ing. Petr Číka, PhD.
- [7] Maxim Integrated, Inc. *What Is an iButton Device?*. [online]. 2006. [cit. 2019-10-19]. Dostupné z URL: <<https://www.maximintegrated.com/en/design/technical-documents/app-notes/3/3808.html>>.  
<https://www.ict.co/Choosing-Card-Technology>
- [8] ICT. *Choosing Card Technology*. [online]. [cit. 2019-10-21]. Dostupné z URL: <<https://www.ict.co/Choosing-Card-Technology>>.
- [9] NXP. *MFRC522*. [online]. 2016. [cit. 2019-10-21]. Dostupné z URL: <<https://www.nxp.com/docs/en/data-sheet/MFRC522.pdf>>.
- [10] Diffen.com. *MMC vs. SD Card*. [online]. 2016. [cit. 2019-10-21]. Dostupné z URL: <[https://www.diffen.com/difference/MMC\\_vs\\_SD\\_Card](https://www.diffen.com/difference/MMC_vs_SD_Card)>.
- [11] Maxim Integrated, Inc. *DS3231*. [online]. 2015. [cit. 2019-10-23]. Dostupné z URL: <<https://datasheets.maximintegrated.com/en/ds/DS3231.pdf>>.

- [12] SOLOMON SYSTECH. *SSD1306*. [online]. 2008. [cit.2019-10-23]. Dostupné z URL: <<https://cdn-shop.adafruit.com/datasheets/SSD1306.pdf>>.
- [13] Toshiba Semiconductor. *TLP350*. [online]. 2017. [cit.2019-10-24]. Dostupné z URL: <<https://toshiba.semicon-storage.com/us/product/opto/photocoupler/detail.TLP350.html>>.
- [14] Texas Instruments. *ULN2803*. [online]. 2017. [cit.2019-10-24]. Dostupné z URL: <<http://www.ti.com/lit/ds/symlink/uln2803a.pdf>>.
- [15] GME. *Indukční čidlo LJ12A3-4-Z/BX, M12, 0-4 mm, NPN/NO, kabel*. [online]. 2019. [cit.2019-10-27]. Dostupné z URL: <<https://www.gme.cz/indukcni-cidlo-lj12a3-4-z-bx-m12-0-4-mm-npn-no-kabel>>.
- [16] Meanwell-web.com *IRM-10-12*. [online]. 2019. [cit.2019-10-27]. Dostupné z URL: <<https://www.meanwell-web.com/en-gb/ac-dc-single-output-encapsulated-power-supply-irm--10--12>>.
- [17] Batteryspace.com *NCR18650B datasheet*. [online]. 2012. [cit.2019-10-28]. Dostupné z URL: <<https://www.batteryspace.com/prod-specs/NCR18650B.pdf>>.
- [18] Linear Technology Corporation. *LTC4415 Dual 4A Ideal Diodes with Adjustable Current Limit*. [online]. 2011. [cit.2019-10-28]. Dostupné z URL: <<https://www.analog.com/media/en/technical-documentation/data-sheets/4415fa.pdf>>.
- [19] Linear Technology Corporation. *Standalone LinearLi-Ion Battery Charger with Thermal Regulation in ThinSOT*. [online]. 2003. [cit.2019-10-28]. Dostupné z URL: <<https://www.analog.com/media/en/technical-documentation/data-sheets/405442xf.pdf>>.
- [20] Linear Technology Corporation. *LTC4412 – Low Loss PowerPat Controller in ThinSOT*. [online]. 2002. [cit.2020-05-20]. Dostupné z URL: <<https://www.analog.com/media/en/technical-documentation/data-sheets/4412fb.pdf>>.
- [21] Espressif Systems *Get Started*. ESP-IDF Programming Guide. [online]. 2016. [cit.2019-11-01]. Dostupné z URL: <<https://docs.espressif.com/projects/esp-idf/en/latest/get-started/index.html>>.
- [22] MQTT.ORG *Documentation–Protocol Specifications*. [online]. 2019. [cit.2019-11-01]. Dostupné z URL: <<http://mqtt.org/documentation>>.

- [23] ON Semiconductor/Fairchild Semiconductor. *FDN306P*. [online]. 2001. [cit. 2020-05-20]. Dostupné z URL: <<https://www.tme.eu/Document/896bf5a5e439b4f373f7fac1cdc11e75/FDN306P.pdf>>.
- [24] Fairchild Semiconductor. *Single-Channel 6N137 High Speed 10MBit/s Logic Gate Optocouplers*. [online]. 2011. [cit. 2020-05-21]. Dostupné z URL: <<http://www.farnell.com/datasheets/1057512.pdf>>.
- [25] SHARP. *PC817X Series DIP 4pin General Purpose Photocoupler*. [online]. 2003. [cit. 2020-05-21]. Dostupné z URL: <<https://www.farnell.com/datasheets/73758.pdf>>.
- [26] retry EDA tools and systems. *České manuály pro Altium Designer*. [online]. 2012. [cit. 2020-05-21]. Dostupné z URL: <<https://edatools.cz/podpora/uvodad>>.
- [27] Microchip Technology Inc. *MCP23017/MCP23S17 16-Bit I/O Expander with Serial Interface*. [online]. 2016. [cit. 2020-05-22]. Dostupné z URL: <<http://ww1.microchip.com/downloads/en/devicedoc/20001952c.pdf>>.

# Zoznam symbolov, veličín a skratiek

<b>AC</b>	striedavý prúd – Alternating Current
<b>ACLU</b>	American Civil Liberties Union
<b>AMQP</b>	Advanced Message Queuing Protocol
<b>CAN</b>	Controller Area Network
<b>CC/CV</b>	Constant Current / Constant Voltage
<b>CoAP</b>	Constrained Application Protocol
<b>CPLD</b>	Complex Programmable Logic Device – zložitý programovateľný logický obvod
<b>CPU</b>	central processing unit
<b>DC</b>	jednosmerný prúd – Direct Current
<b>DSP</b>	doska plošných spojov
<b>DRC</b>	design rule checking
<b>EEPROM</b>	Electrically Erasable Programmable Read-Only Memory
<b>EDL</b>	Eclipse Distribution License
<b>EPL</b>	Eclipse Public License
<b>EPS</b>	Evolved Packet System
<b>FPGA</b>	Field programmable gate array – pole logických členov programovateľné užívateľom
<b>GPIO</b>	General-purpose input/output – univerzálny vstupno/výstupný pin
<b>GUI</b>	Graphical User Interface – grafické používateľské rozhranie
<b>HTTP</b>	hypertextový prenosový protokol – HyperText Transfer Protocol
<b>I<sup>2</sup>C</b>	meno štandardu sériovej zbernice – Inter-Integrated Circuit
<b>IDE</b>	vývojové prostredie – Integrated Development Environment
<b>I/O</b>	vstupno-výstupný – Input/Output
<b>IoT</b>	Internet of Things – internet vecí
<b>IP</b>	Internet Protocol
<b>IPv6</b>	Internet Protocol version 6
<b>JSON</b>	JavaScript Object Notation
<b>LSB</b>	Least Significant Bit
<b>LTE</b>	Long Term Evolution
<b>M2M</b>	Machine to Machine
<b>MIFA</b>	Meandered inverted-F antenna
<b>MMC</b>	MultiMediaCard
<b>MSB</b>	Most Significant Bit
<b>MQTT</b>	Message Queuing Telemetry Transport
<b>NVRAM</b>	Non-Volatile Random Access Memory – energeticky nezávislá pamäť s priamym prístupom

<b>PHP</b>	Hypertext Preprocessor
<b>PWM</b>	pulzná šírková modulácia – Pulse Width Modulation
<b>RAM</b>	Random Access Memory
<b>RFID</b>	Radio Frequency IDentification
<b>RISC</b>	Reduced instruction set computer – počítač s obmedzenou sadou inštrukcií
<b>RTC</b>	hodiny reálneho času – Real Time Clock
<b>RX</b>	prijat – Receive
<b>SAR</b>	Succesive Approximation Register
<b>SD</b>	Secure Digital
<b>SMD</b>	Surface Mounted Device
<b>SMT</b>	Surface Mounted Technology
<b>SPI</b>	sériové periférne rozhranie – Serial Peripheral Interface
<b>SRAM</b>	Static Random Access Memory
<b>SQL</b>	Structured Query Language
<b>THT</b>	through hole technology
<b>TX</b>	odoslať – Transmit
<b>UART</b>	univerzálny asynchrónny prijímač / vysielateľ – Universal Asynchronous Receiver-Transmitter
<b>USB</b>	univerzálna sériová zbernica – Universal Serial Bus
<b>UVLO</b>	UnderVoltage LockOut – podpäťové blokovanie
<b>WIFI</b>	wireless fidelity
<b>XML</b>	eXtensible Markup Language

$A$	Ampér
$mA$	miliAmpér
$\mu A$	mikroAmpér
$dbm$	decibel to miliwatt
$MHz$	megaHertz
$kHz$	kiloHertz
$Hz$	Hertz
$kbit/s$	kilobit za sekundu
$Mbit/s$	megabit za sekundu
$\Omega$	Ohm
$nF$	nanoFarad
$\mu F$	mikroFarad
$V$	Volt
$mV$	miliVolt
$W$	Watt

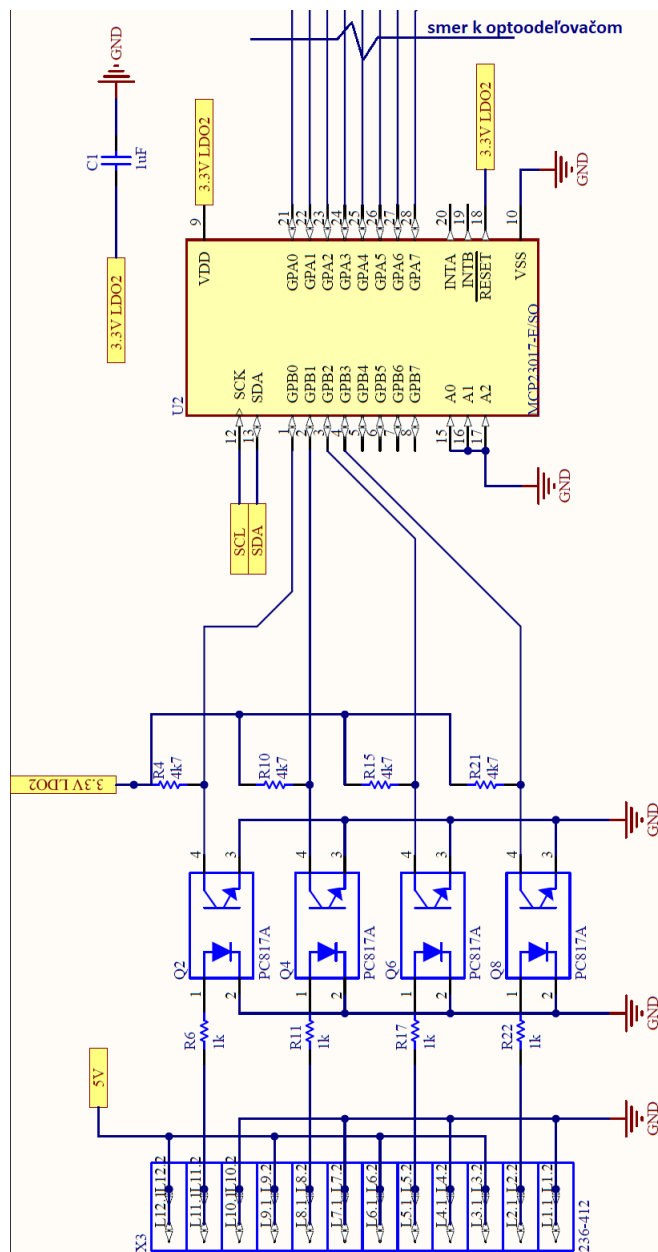


# Zoznam príloh

<b>A</b>	<b>Schémy</b>	<b>81</b>
A.1	Schéma vstupov a výstupov MPC23017 1/2 . . . . .	81
A.2	Schéma vstupov a výstupov MPC23017 2/2 . . . . .	82
A.3	Napájanie/regulátory . . . . .	83
A.4	Aktualizovaný obvod nabíjania . . . . .	84
A.5	Obvod prepínania napájania s LTC4412 . . . . .	85
<b>B</b>	<b>3D náhľady</b>	<b>86</b>
B.1	Doska s 3D modelmi . . . . .	86
B.2	Doska bez 3D modelov . . . . .	87
B.3	Bočný 3D pohľad . . . . .	88
B.4	Náhľad na medenú vrstvu . . . . .	89
B.5	Pohľad do dashboardu . . . . .	90

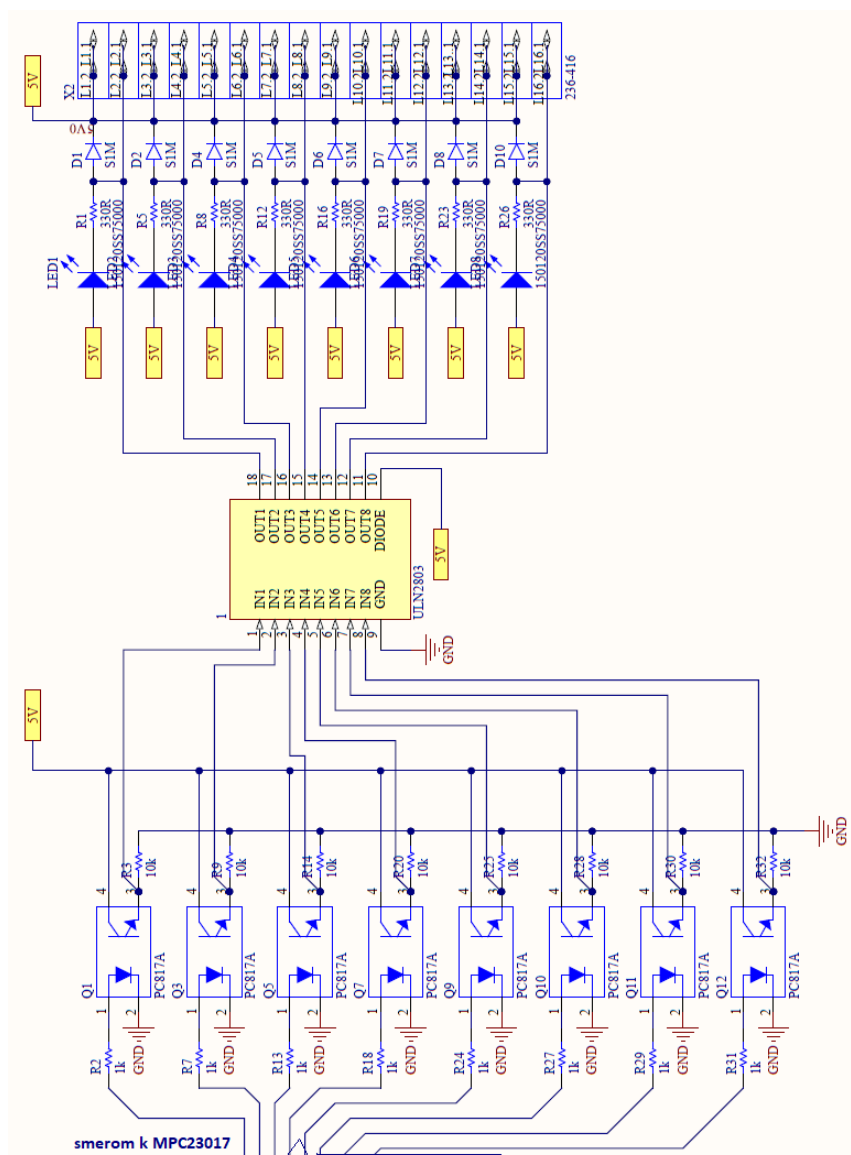
## A Schémy

### A.1 Schéma vstupov a výstupov MPC23017 1/2



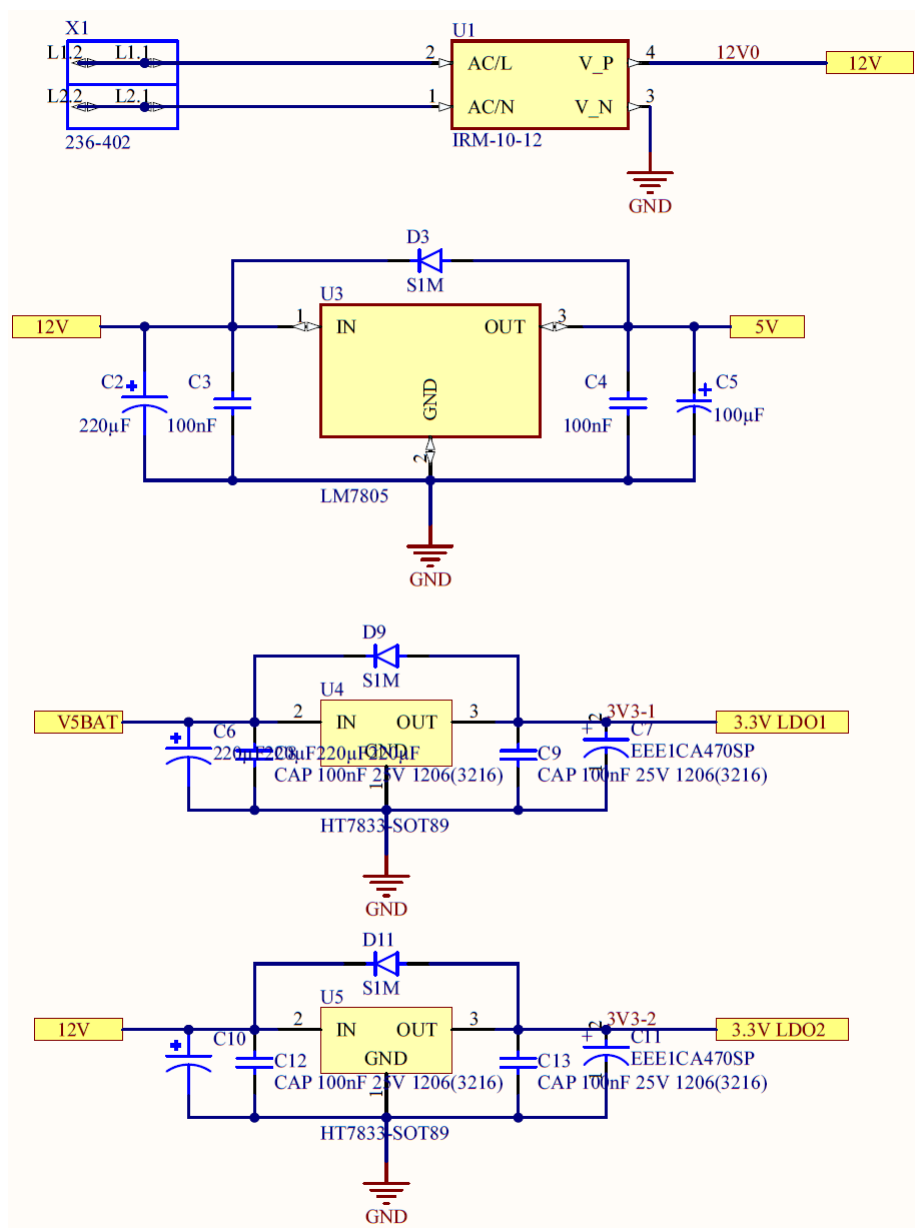
Obr. A.1: Prvá časť schémy vstupov a výstupov MPC23017

## A.2 Schéma vstupov a výstupov MPC23017 2/2



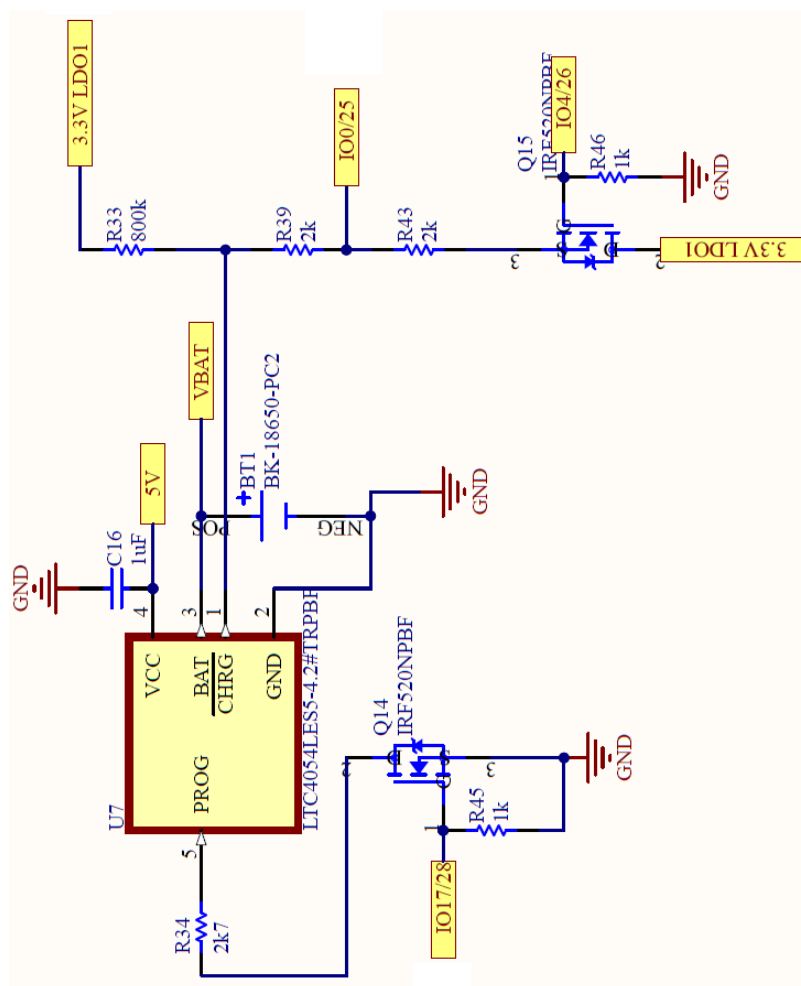
Obr. A.2: Druhá časť schémy vstupov a výstupov MPC23017

## A.3 Napájanie/regulátory



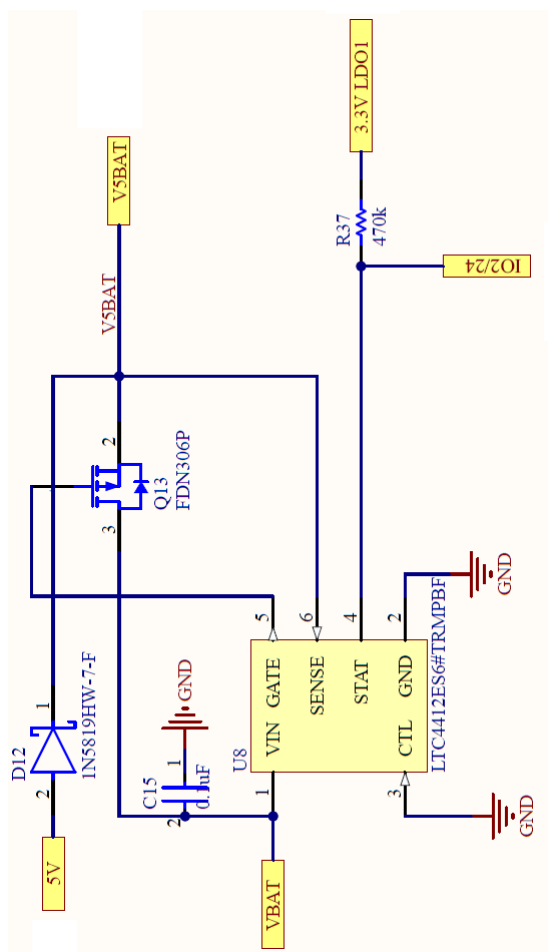
Obr. A.3: Vstup napájania a regulátory napätia

## A.4 Aktualizovaný obvod nabíjania



Obr. A.4: Schéma obvodu nabíjania a detekcie stavov nabíjačky

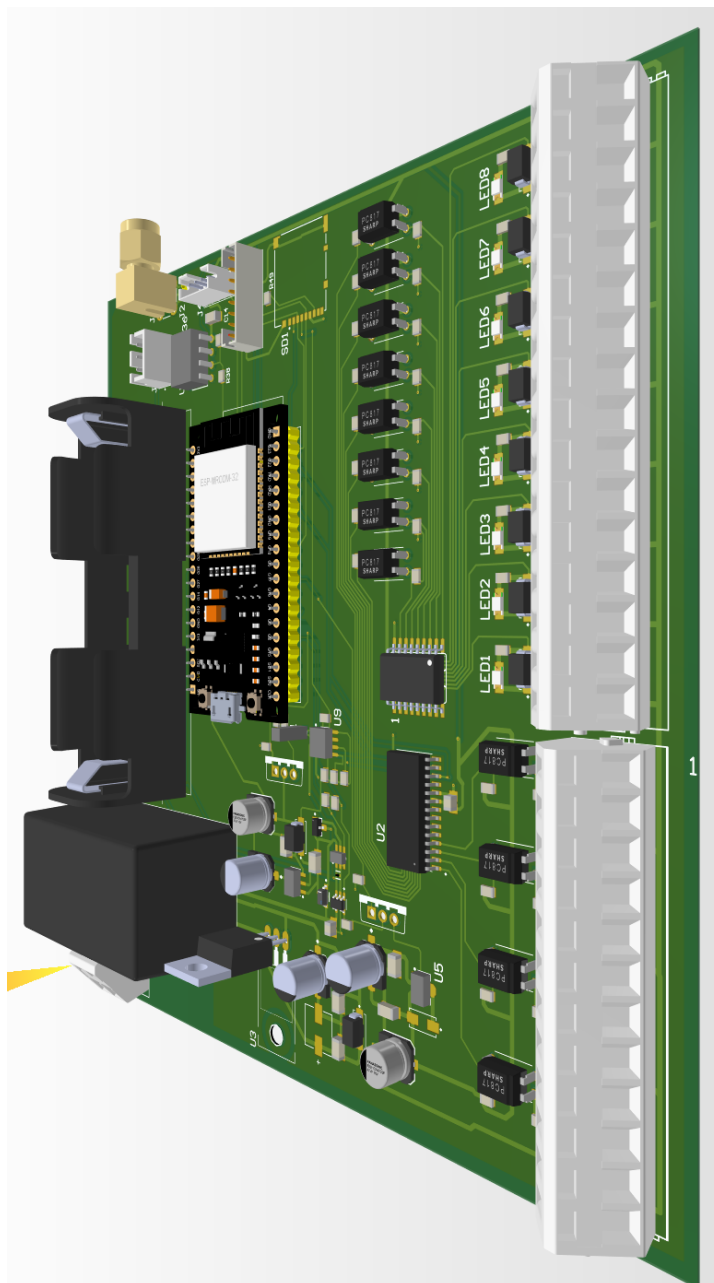
## A.5 Obvod prepínania napájania s LTC4412



Obr. A.5: Schéma obvodu prepínania napájania

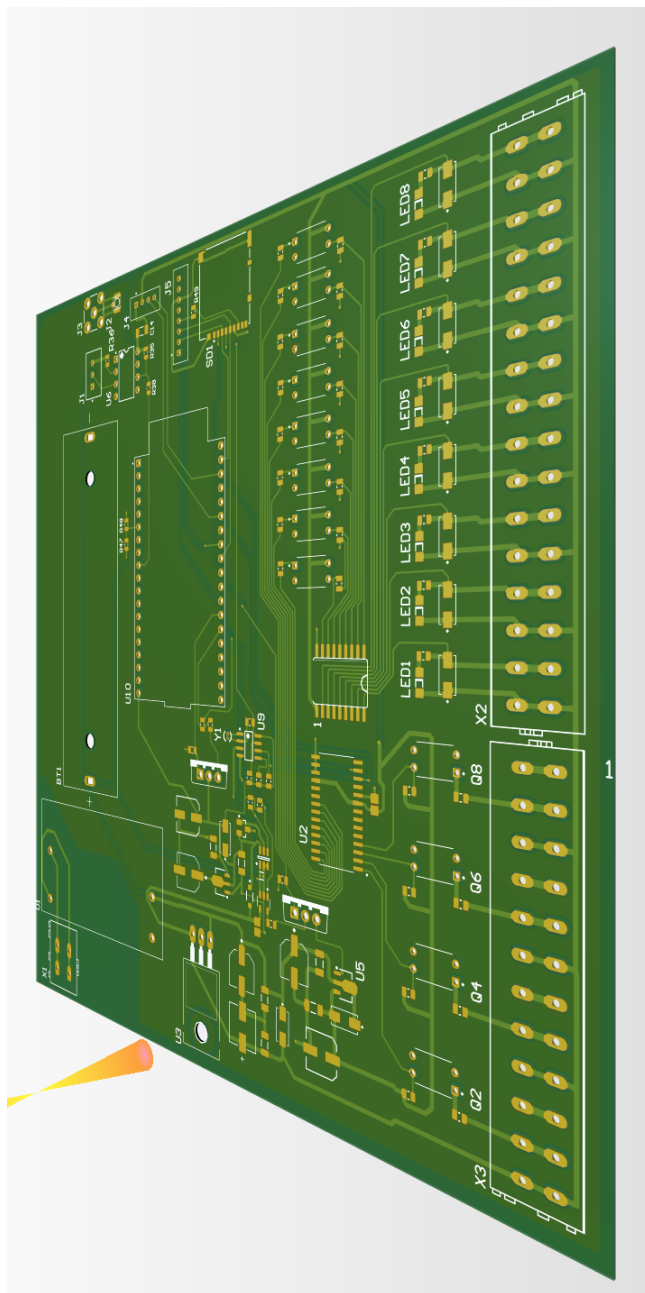
## B 3D náhľady

### B.1 Doska s 3D modelmi



Obr. B.1: Pohľad na 3D verziu so zapnutými modelmi

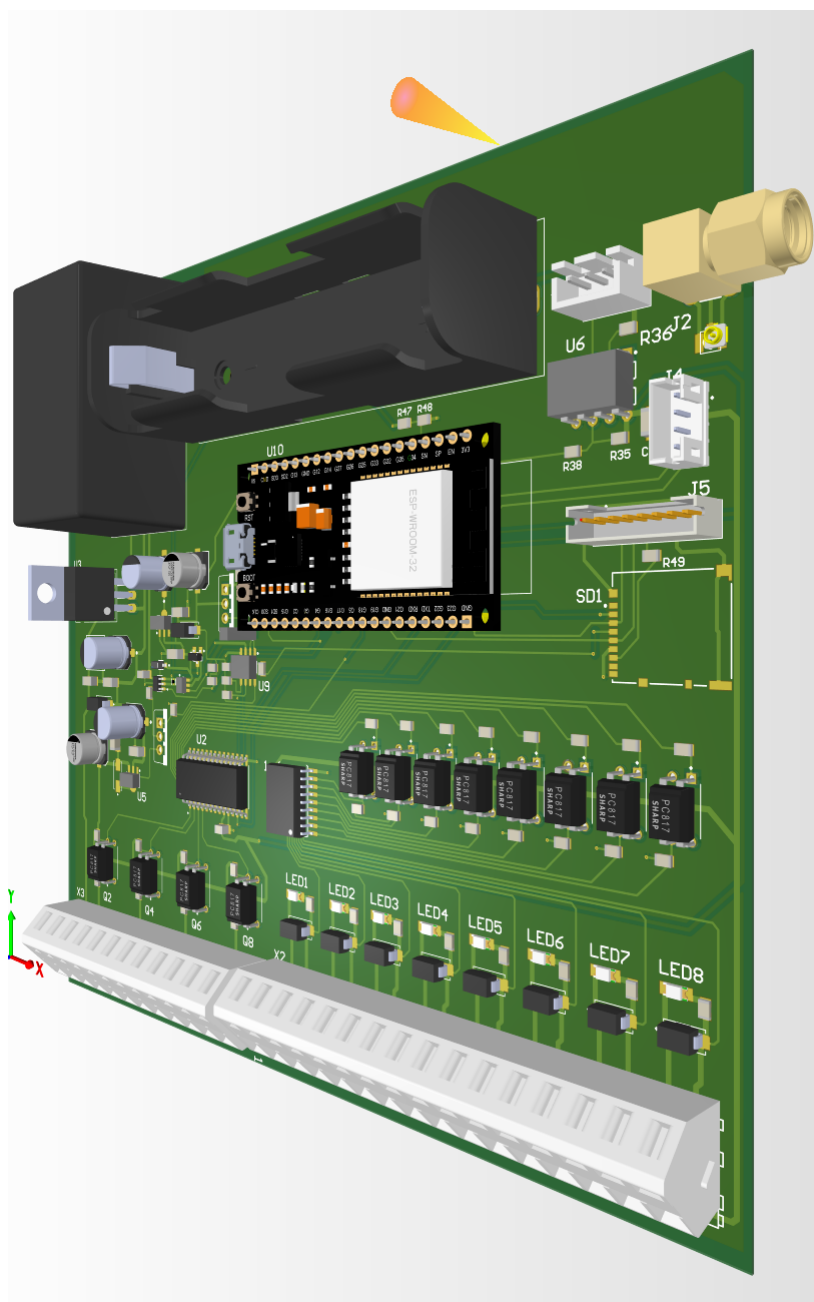
## B.2 Doska bez 3D modelov



Obr. B.2: Pohľad na 3D verziu s vypnutými modelmi

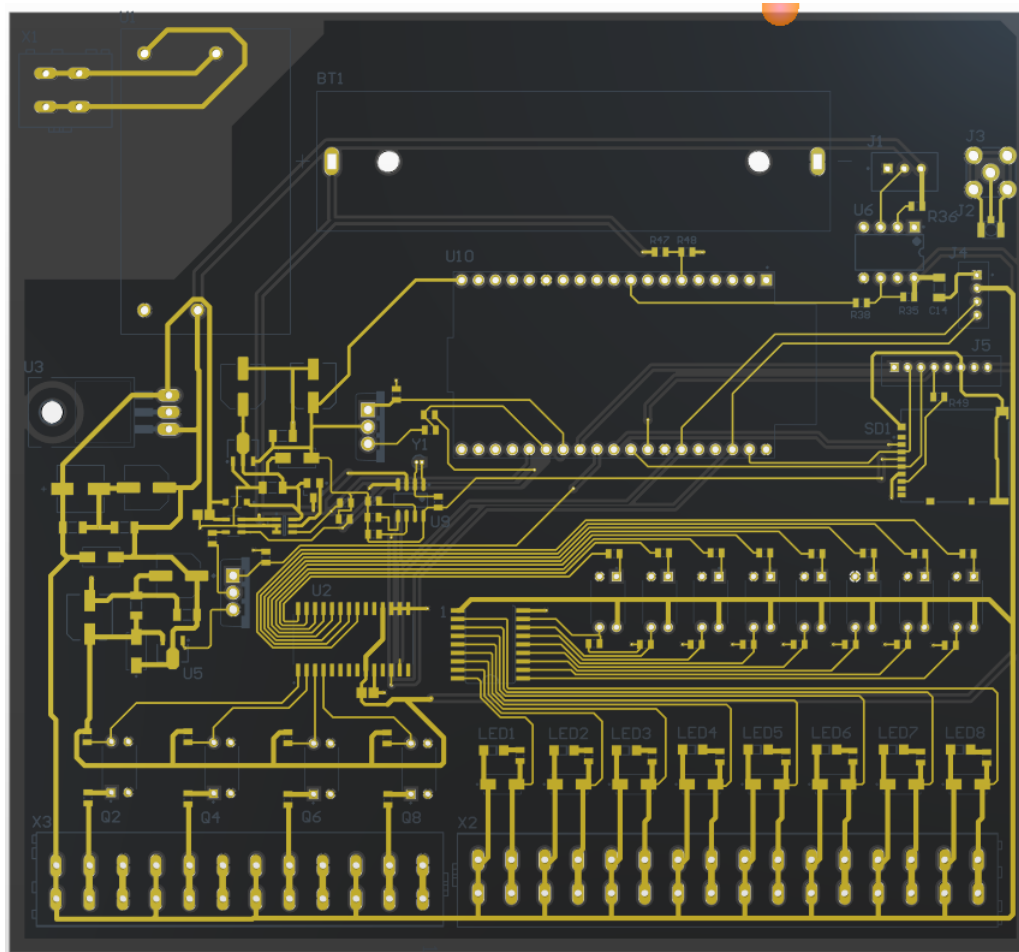


### B.3 Bočný 3D pohľad



Obr. B.3: Bočný 3D pohľad so zapnutými modelmi

## B.4 Náhľad na medenú vrstvu



Obr. B.4: V 3D zvýraznenie spojov

## B.5 Pohľad do dashboardu



Obr. B.5: Aktívny dashboard